ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

4

Cómo elegir una Base de Datos

Aula de Informática

SINCLAIR
AMSTRAD
COMMODORE
IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

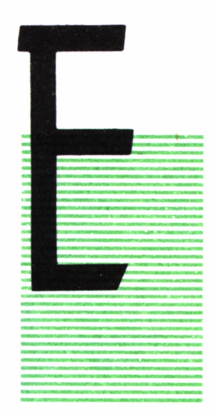
INFORMATICA APLICADA

4

Cómo elegir una Base de Datos

EDICIONES SIGLO CULTURAL

INTRODUCCION 1



EL desarrollo, estandarización y abaratamiento de los ordenadores personales ha hecho posible la extensión y difusión de procedimientos automatizados en áreas de actividad en las que el empleo de esos procedimientos quedaba reservado, por su complejidad y elevado coste, a las grandes empresas e instituciones y a la Administración pública. Desde hace media década existe un producto software potente para microordenadores, que se ha dado en llamar «base de datos».

Tradicionalmente, en Informática se ha denominado «fichero» a un conjunto de datos homogéneos, y «banco de datos» a toda colección de distintos ficheros dedicados a una misma aplicación. Muchos usuarios comenzaron a utilizar la expresión «bases de datos» para designar indistintamente a ficheros y bancos y datos. No cabe duda, pues, antes de entrar en materia, que es preciso clarificar el significado real de base de datos.



¿QUE ES UNA BASE DE DATOS?

En primera instancia, cabe afirmar rotundamente que una base de datos no es un fichero y tampoco un banco de datos. Una definición clásica de base de datos puede ser la siguiente: una base de datos es un conjunto de datos relacionados entre sí, que se encuentran almacenados en una única colección, sin redundancias, y que cumple las siguientes condiciones:

1. Los datos se encuentran almacenados en diversos soportes de información, de tal forma que son independientes de los programas que los controlan.
2. Su utilización no está restringida a una aplicación exclusiva, siendo posible su acceso por parte de varias aplicaciones, incluso simultáneamente.

3. Sirven para gestionar la información contenida en la base de datos, esto es, para incluir nuevos datos, borrar datos especialmente diseñados para optimizar el funcionamiento del Sistema. Estos procedimientos no sólo se encargan de la actualización, sino que también permiten la obtención de datos para su uso o mediante programas.

Al resumir la anterior definición formal, destacan dos conceptos fundamentales:

1. Una base de datos es información almacenada en soportes.
2. Una base de datos es una serie de procedimientos (programas) encargados de gestionar información.

Esta doble faceta es la que ha dado lugar a innumerables confusiones. Por un lado, algunos asocian base de datos con información, mientras que otros asocian dicho concepto con los programas para la gestión de la información. En general, este último concepto se suele diferenciar llamándolo Sistema de Gestión de la Base de Datos (SGBD), distinguiéndolo de los datos propiamente dichos.



Fig. 1.1.



PRECEDENTES

En sus comienzos, las bases de datos fueron conocidas como «sistemas de información de gestión». La idea era poner todos los datos de las compañías en una gran «piscina» en donde los ejecutivos pudieran «ir de pesca» de cualquier información que desearan. La información reflejaría instantáneamente, por supuesto, cualquier cambio resultante de nóminas de pago, facturas de ventas, etc. Pero este sistema no era adecuado porque el software no era lo suficientemente sofisticado. Después de fallar en el intento de implementación de los sistemas de información de gestión, los profesionales de las computadoras reconsideraron y aplicaron el concepto de base de datos a conjuntos específicos de operaciones sobre los datos.

El origen de las bases de datos es una consecuencia en sí misma de la evolución de la Informática en los últimos años.

Cuando sólo existía la monoprogramación, los ficheros no eran compartidos por distintas aplicaciones, por lo cual la integridad y seguridad de los datos no comportaba problemas.

Con la multiprogramación, al permitirse que varias aplicaciones se ejecuten a un mismo tiempo, surge un conflicto de integridad entre los ficheros compartidos.

Entre las características de los sistemas de gestión de base de datos, destaca el que están dotados de una serie de controles y seguridades que evitan que se dé este problema.

Desde que el uso de microordenadores se ha generalizado, han aparecido numerosos programas «a medida» para tratar el tipo de información descrita anteriormente. La posibilidad de realizar una mecanización con éxito reside, sin embargo, en la selección del programa adecuado.

La creciente disponibilidad en el mercado de paquetes de software para las bases de datos ha vuelto totalmente antieconómica la alternativa de desarrollar paquetes a medida, planteando, sin embargo, el problema de seleccionar el paquete de software más adecuado a nuestras necesidades.



GESTION FICHEROS VS. BASES DE DATOS

En el mercado de programas para microordenadores hay ciertos programas cuyos creadores denominan «bases de datos». Muchos de ellos son simples gestores de ficheros que solamente sirven para abrir un fichero de datos y realizar informes y no suelen tener lenguaje de programación.



¿POR QUE LAS BASES DE DATOS?

Los sistemas informáticos tradicionales han sido llamados también sistemas orientados hacia el proceso, debido a que en ellos se pone el acento en el tratamiento que reciben los datos, los cuales se almacenan en unos ficheros, que son usados para una única aplicación. Las aplicaciones se analizan e implantan con entera independencia unas de otras, y los datos no se suelen transferir entre ellas, sino que se duplican siempre que las correspondientes aplicaciones tengan necesidad de ellos.

Esto produce, además de una ocupación inútil de memoria, un aumento en los tiempos de proceso, repitiendo los mismos controles y operaciones en los distintos ficheros. Pero, más grave todavía, son las incoherencias que se presentan cuando la actualización de los datos, que se incluyen en más de un fichero, no se realiza de forma simultánea.

Por otra parte, la dependencia de los datos respecto al soporte físico y a los programas da lugar a una falta de flexibilidad y de adaptabilidad frente a los cambios, lo cual repercute muy negativamente en el rendimiento del conjunto del sistema informático.

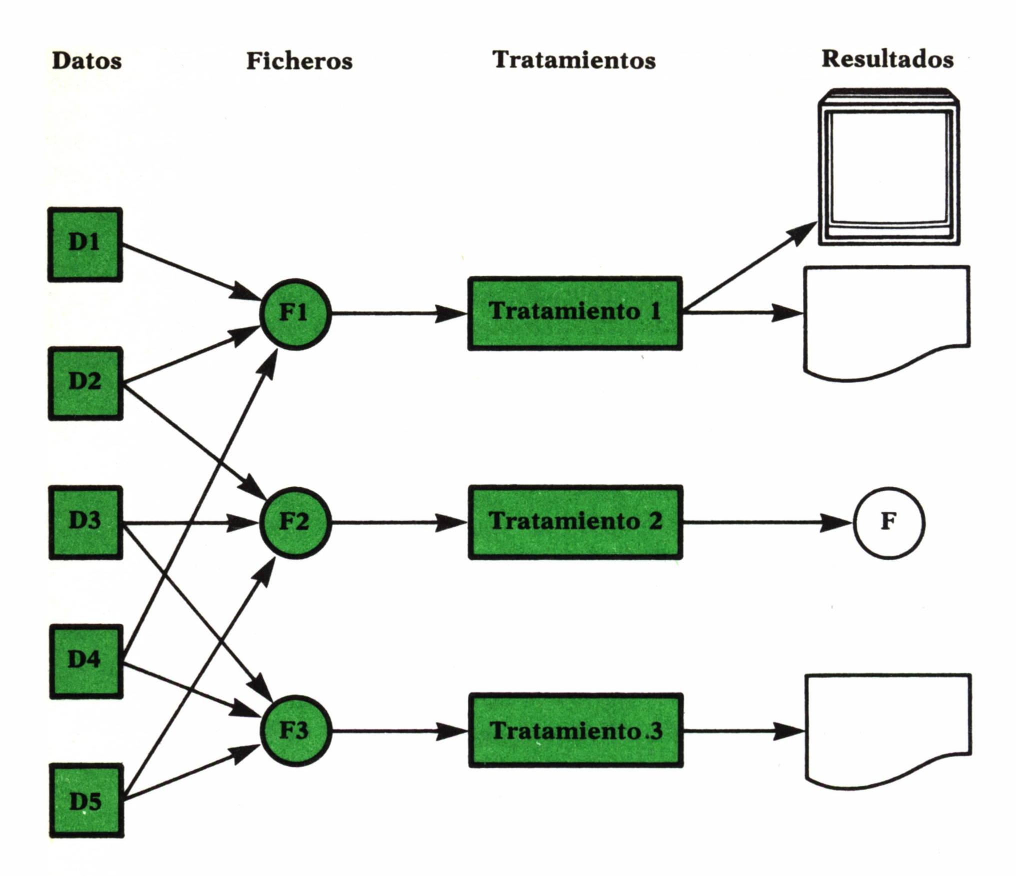


Fig. 1.2. Organización clásica. Cada tratamiento tiene su fichero específico, existiendo una gran redundancia en los datos.

La solución a estos problemas lleva a adoptar un punto de vista radicalmente distinto del considerado en los sistemas tradicionales, pasando los tratamientos a ocupar un segundo plano, mientras que los datos adquieren el protagonismo.

Los datos se organizan y se mantienen en un conjunto estructurado, el cual no está diseñado para una aplicación concreta, sino que, por el contrario, tienden a satisfacer las necesidades de información global del usuario.

Estos sistemas orientados hacia los datos van sustituyendo a los sistemas orientados hacia el proceso que, por su poca fiabilidad, falta de representación de la realidad y mal asegurada confidencialidad van perdiendo la confianza del usuario.

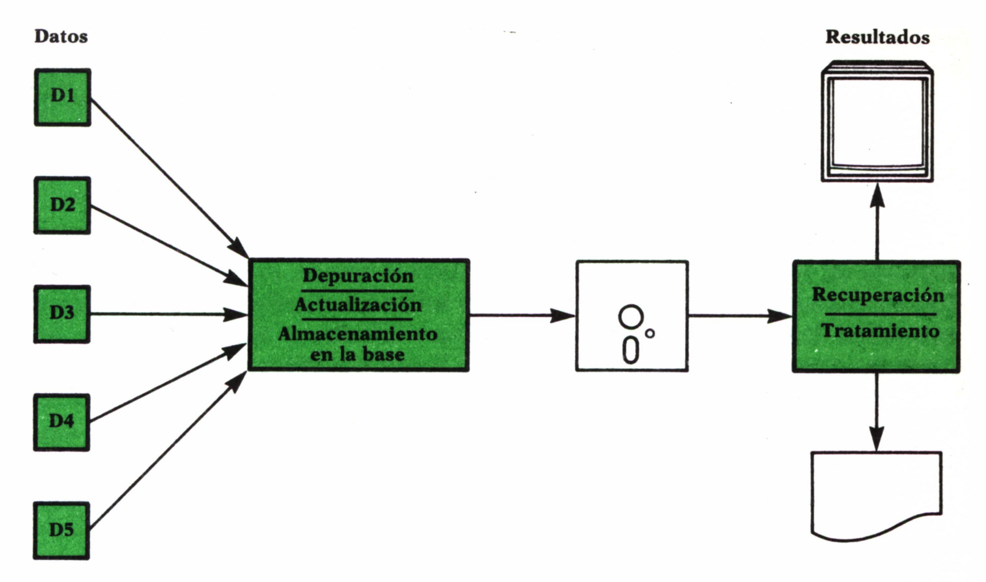


Fig. 1.3. Organización en base de datos. Los datos se recogen y almacenan con independencia de los tratamientos a que van a ser sometidos. No hay redundancias.



VENTAJAS DE LAS BASES DE DATOS

La aplicación de la metodología de bases de datos puede prestar un gran servicio a fin de conseguir un sistema de información coherente y que proporcione el soporte para la toma de decisiones, presentando una multitud de ventajas frente a los sistemas clásicos:

a) *Mayor valor informativo.* Puesto que la base de datos es un reflejo del mundo real, donde los distintos elementos están interrelacionados, el valor informativo de su conjunto es superior a la suma del valor informativo de los elementos individuales que los constituyen. Por ejemplo, una base de datos que contuviera los datos referentes a proveedores y productos tendrá un valor informativo mayor que la lista de proveedores y la lista de productos de forma independiente.

b) *Independencia de los datos respecto a los tratamientos, y viceversa.* La mutua independencia de datos y tratamientos lleva a que un cambio de estos últimos no imponga una nueva estructuración de la base; y, por otra parte, la inclusión de nuevas informaciones en la base, desaparición de obras, etc., no debe obligar a alterar los programas. Esta independencia de los tratamientos frente a la estructura de base supone una considerable ventaja, al evitar el importante esfuerzo que origina la reprogramación de las aplicaciones cuando se producen cambios en los datos (en algunos

equipos se dedica cerca del 50% de los efectivos de programación al mantenimiento de los sistemas existentes).

c) *Coherencia de los resultados.* Debido a que la información de la base es recogida y almacenada una sola vez, en todos los tratamientos se utilizan los mismos datos, por lo que los resultados de todos ellos son coherentes y perfectamente comparables.

Además, al no existir, o al menos, disminuir en gran medida las redundancias en los datos, desaparece el problema que se presentaba en el enfoque clásico, de que el cambio de un mismo dato obligaba a actualizar una serie de ficheros; y con él, se elimina también el inconveniente de las divergencias en los resultados debidos a actualizaciones no simultáneas en todos los ficheros.

d) *Mejor disponibilidad de los datos para el conjunto de usuarios.* Cuando se aplica la metodología de bases de datos, cada servicio ya no es «propietario» de los datos, puesto que éstos se comparten entre el conjunto de los usuarios, existiendo una mejor disponibilidad de los datos para todos los que tienen necesidad de ellos, siempre que estén autorizados para accederlos.

Hay, también, una mayor «transparencia» respecto a la información existente, ya que todos los datos que se encuentren en la base se deben relacionar en un catálogo que puede ser ampliamente difundido e, incluso, en bastantes casos, accedido por medios informáticos.

e) *Mejor y más normalizada documentación de la información, la cual esté integrada con los datos.* En el enfoque clásico, los datos se encuentran separados de su contenido semántico; los primeros se almacenan en ficheros y su descripción se hace mediante un lenguaje de programación. La documentación de los datos en soporte de papel, realizada por el analista o programador, es por lo general insuficiente y a veces incluso inexistente; además, por lo común la normalización brilla por su ausencia. Este problema se atenúa en gran medida con las bases de datos, ya que en la misma base se incluye no sólo la información, sino también la documentación de éste, la cual, por tanto, está normalizada.

f) *Mayor eficiencia en la recogida, codificación y entrada de los datos al sistema.* Al no existir apenas redundancias, y ser los datos recogidos una sola vez, aumenta el rendimiento de todo proceso previo al almacenamiento.

g) *Reducción del espacio de almacenamiento.* La desaparición, o disminución de las redundancias, así como la aplicación de las técnicas de compresión, lleva, en los sistemas de bases de datos, a una menor ocupación del almacenamiento secundario —en general, pequeño—; sin embargo, los elementos del Sistema ocupan a veces tanto espacio que el ahorro es más técnico que práctico.

La aplicación de la metodología de bases de datos puede prestar un gran servicio a fin de conseguir sistemas de información coherentes y que proporcionen el debido soporte para la toma de decisiones.



TERMINOLOGIA

Los términos utilizados en la industria del software se suelen originar en los Manuales de usuario. Como se podría esperar, hay poco acuerdo sobre el término de base de datos. Esta sección establece cómo serán utilizados los términos de este libro. Estos se emplearán de forma coherente, aun cuando proveedores particulares describan sus sistemas con otros términos. Si al principio estas definiciones parecen algo confusas, no hay por qué preocuparse; serán comprendidos a medida que se lea el libro.



Conceptos fundamentales

Las denominaciones de fichero y archivo utilizadas en Informática provienen de la Gestión Administrativa de la Empresa. En el campo administrativo, un fichero es el conjunto de información almacenado en cartulinas o fichas; en cambio, en Informática, este conjunto de información estaría codificado en un soporte no legible por el hombre. Esto ha abierto, en la mayoría de los casos, enormes posibilidades al permitir almacenar grandes volúmenes de información que pueden ser consultados en un tiempo muy breve.

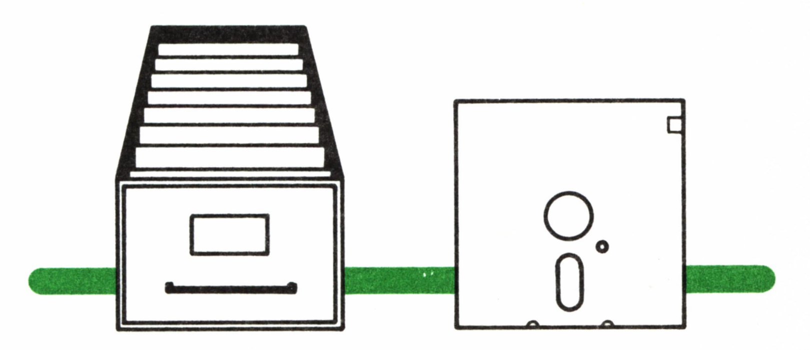


Fig. 1.4.



FICHERO

Se denomina fichero al conjunto de elementos que contienen información relativa a un tema común, estando ésta de acuerdo con unos forma-

tos preestablecidos. De forma más elemental diríamos que un fichero es un conjunto de «registros lógicos» de la misma naturaleza, por ejemplo, el fichero de clientes, fichero de proveedores, fichero de cuentas, etc.



REGISTRO Y CAMPO

Llamamos *«registro»* o «registro lógico» al conjunto de datos referidos a un mismo elemento. Por ejemplo: si tomamos como elemento uno de los empleados de una determinada empresa, el conjunto de datos referidos a dicho empleado constituirán lo que denominaríamos «registro». Así, podemos incluir como datos en este registro: número de empleado, nombre, dirección, categoría, antigüedad, etc.

Como hemos visto, dentro de un registro hay diversas informaciones; cada una de ellas será un *«campo»* del registro. A su vez, el campo se subdivide en *«subcampos»*. En el ejemplo anterior el registro «empleado» consta de los campos: número de empleado, nombre, dirección, etc., mientras que el campo «dirección» consta de los subcampos: domicilio y localidad.

N.° Empleado	Nombre			Dirección		Categoría	Antigüedad
	Nombre-P	Apell-1	Apell-2	Domicilio	Antigüedad		
1001	J. María	García	Pérez	Alcalá, 50	Madrid	0045	003

1 4 51 78 91 95 98

Longitud del registro

Fig. 1.5.

Los campos pueden ser:

— *Numéricos,* si sólo contienen números.
— *Alfabéticos,* únicamente contienen letras.
— *Alfanuméricos,* cuando contienen números, letras y caracteres especiales.
— El dato N. EMPLEADO 1001 está contenido en el campo N. EMPLEADO.
— El dato JOSE MARIA está contenido en el subcampo NOMBRE-P.



CLAVES E INDICES

Entre los datos de un registro suele haber uno de ellos que sirve para identificarle. Su principal característica es la de no repetirse a lo largo del

fichero. A este dato lo llamaremos *CLAVE*. En nuestro caso es el número de empleados dentro de la empresa.

Los datos ocupan posiciones dentro del registro, llamados también **CARACTERES**. Un carácter es la unidad más elemental dentro del registro.

INDICE. Un índice es una tabla de doble entrada; es decir, una lista de claves que se corresponden con un número de registro. Por ejemplo, todo libro tiene su índice temático, en el cual para cada tema tenemos un número de página. Los números de registro se llaman «punteros» y están dispuestos para ayudar a encontrar un registro particular mediante una clave; al igual que el índice de un libro, nos permite encontrar rápidamente el capítulo buscado.



DICCIONARIO DE DATOS

El diccionario de datos es una descripción completa de los campos en una base de datos. El diccionario de datos describe las relaciones entre los diversos campos en una base de datos. También describe cada campo por nombre, membrete de encabezamiento de informe, la extensión, tipo de datos, etc.



Base de datos

Una base de datos es una colección de datos interrelacionados. Concretamente, la base de datos es una colección completa de datos, punteros, tablas, índices, direcciones, etc. Si mantenemos el registro de sus clientes y de sus productos, ambos forman parte de la misma base de datos, al menos que nunca intentemos relacionar los dos de alguna forma.



SISTEMA DE GESTION DE BASE DE DATOS

Un SGBD es un paquete de programas de computadora y documentación que nos permite instalar y utilizar una base de datos.



ACCESO DE FICHEROS

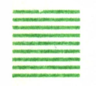
Por «acceso a un fichero» entendemos el modo utilizado para la lectura de los registros del mismo. Los modos de acceso pueden ser alguno de los siguientes:

SECUENCIAL. Para acceder a un determinado registro, se van leyendo todos los registros que componen el fichero según la secuencia de alma-

cenamiento, hasta localizar el buscado, que en particular puede ser el registro de fin de fichero.

DIRECTO O ALEATORIO. A un registro se accede mediante su clave, sin necesidad de leer los registros que le preceden.

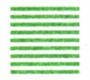
DINAMICO. Este método permite acceder a determinadas franjas de registros del fichero claramente delimitados, tanto directa como secuencialmente.



OPERACIONES POSIBLES SOBRE FICHEROS

En esta sección desarrollaremos la terminología que utilizaremos para realizar el tratamiento deseado sobre nuestros ficheros intentando clarificar la diferencia entre cada una de las diferentes funciones.

Hemos establecido anteriormente que un SGBD es un conjunto de programas de ordenador que podemos contemplar como «programa integrado». Una serie de órdenes o comandos, a introducir a través del teclado, permite comunicar al programa, en cualquier momento, las funciones que el usuario quiere ejecutar.



CREACION

Una de las funciones principales de un programa de esta naturaleza es la de permitir fácilmente la especificación («creación») de la estructura del fichero.

Para las ilustraciones que siguen haremos uso de los comandos típicos del «dBaseII», que es uno de los paquetes de software más populares en esta categoría.

La creación de nuestro fichero CATALOGO, de una vez por todas, se realiza comunicando al sistema la siguiente sucesión de comandos. Por supuesto, es preciso haber introducido antes en el ordenador personal el disco correspondiente, que contiene el programa, y haber procedido a inicializar el sistema.

La secuencia de comandos es como sigue:

```
CREATE "CATALOGO"
CODIGOARTICULO, G, C
DESCRIPCION, 20, C
UNIDAD, 20, C
PRECIO, 5, N
```

En resumen, con el primer comando se procedió a comunicar al sistema el nombre del fichero, y con los sucesivos, los nombres y características de los diversos campos, o lo que es lo mismo, la estructura del registro.



INTRODUCCION DE LOS DATOS

Como consecuencia de lo anterior, ya estamos en condiciones, mediante el comando APPEND, de introducir en el fichero nuestros datos a través del teclado.

Para facilitar la tarea del usuario en esta fase aparece en pantalla el formato del registro.

a) Récord 001



b) Récord 001

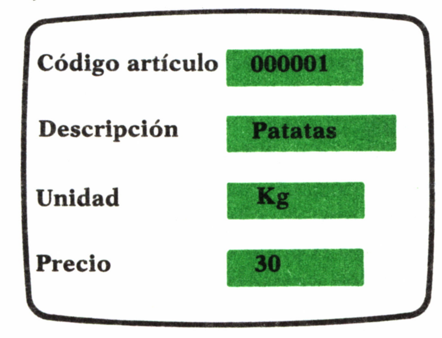


Fig. a) Formato del registro creado para el fichero CATALOGO.
b) Datos introducidos en el primero.



RECUPERACION DE LA INFORMACION

En el supuesto de haber creado el archivo y haber introducido los datos necesarios, podemos proceder a su uso para la obtención de información.

Un archivo real, para que merezca la pena ser controlado por un ordenador, deberá comprender como mínimo un centenar de registros aproximadamente. Definiremos un miniarchivo a título de ejemplo:

Código artículo	Descripción	Unidad	Precio
000001	PATATAS	KG	35
000002	TELAS	MT	150
000003	AZUCAR	KG	90
000004	VINO	LT	100
000005	ACEITE	LT	200

Supongamos que buscamos aquellos artículos que se miden en kilogramos y tienen un precio inferior a 50 pesetas. Los 2 comandos siguientes son suficientes para resolver el problema:

```
USE "CATALOGO"
DISPLAY ALL FOR UNIDAD = "KG" AND PRECIO < "50"
```

El primer comando comunica al sistema la intención del usuario de utilizar el fichero «CATALOGO». El segundo activa la función de solicitud y visualiza en la pantalla todos los registros relativos a los artículos que se miden en kg y tienen un precio inferior a 50 pesetas.

Hay que destacar el hecho de que un programa adecuado hubiera necesitado más de dos instrucciones.

La función descrita y los comandos utilizados nos proporcionan el punto de partida para algunas consideraciones de interés.

Las diversas funciones que una base de datos es capaz de desarrollar son activadas por el usuario mediante una o más órdenes que contienen inicialmente un verbo inglés, seguido por otros elementos específicos. El verbo indica, por lo general, la función a activar, mientras que lo que le sigue tiende a precisar y delimitar el significado del verbo. Así, se consigue que cada comando tenga una estructura formal propia que el usuario deberá conocer a priori para poder ejecutar de forma correcta el propio

comando. Dicho de otro modo, tendrá que conocer el lenguaje con el que hacer funcionar la base de datos.

Desde el punto de vista conceptual, esta circunstancia no es muy diferente del conocimiento de un lenguaje de programación. El aprendizaje del lenguaje de una base de datos, con respecto al de un lenguaje de programación, es más fluido y motivante por las razones siguientes:

— Los comandos que han de aprenderse en el estudio de una base de datos son menos numerosos que los que un lenguaje de programación, al menos en el sentido de que con el conocimiento de un número limitado de comandos, los más importantes, se pueden conseguir resultados significativos.

— No es necesario aprender a estructurar el programa, es decir, a elaborar el diagrama de bloques lógico del programa.

ACTUALIZACION Y MANTENIMIENTO DE LOS DATOS

Una vez que tenemos creado un fichero con sus datos quizá deseemos modificar su estructura de alguna manera, bien modificando las características concretas de los datos en sí mismos, procediendo a dar «altas», «bajas» o «modificaciones» de los registros de nuestra base de datos.

En general, todas las bases de datos proporcionan facilidades que hacen flexible su definición y manipulación, dependiendo en gran medida del sistema que utilicemos.

APLICACIONES PRACTICAS 2



DADO el alto número de bases de datos existentes en el mercado, y la flexibilidad de las mismas, las aplicaciones que podemos hacer con ellas son muchas y muy diversas. El tipo de aplicación que realicemos con cada una dependerá, por un lado, de la capacidad de nuestro equipo y, por otro, de la complejidad de la base. Cuanto más potente sea la base de datos más aplicaciones tendrá. En cualquier caso, teniendo claro desde un primer momento qué es lo que se pretende hacer con la base de datos, y conociendo las restricciones que nos impone la misma, podemos sacarle siempre un buen partido.

Algunos de los posibles usuarios de una base de datos podrían ser:

— Abogados
— Agencias de viajes
— Almacenes
— Bancos
— Farmacias
— Hospitales y clínicas
— Librerías
— Médicos, dentistas, veterinarios
— Seguros
— Vídeo clubes
— Clubes sociales
— Institutos de investigación científica
— Centros informativos o divulgativos
— Bibliotecas
— Comercios
— Editoriales
— Laboratorios

— Universidades
— Servicios de documentación de empresas

En líneas generales, las posibilidades que nos permite una base de datos para una aplicación en concreto son las siguientes:

— Mantener y poner al día el contenido de la base. Entenderemos por mantenimiento la adición de nuevos datos, la modificación de todo o parte del contenido y la anulación de algunos elementos. Si la base de datos es relacional, los nuevos registros pueden ser insertados y cualquier parte de un elemento puede ser modificada o borrada.
— Localización y recuperación de aquellos datos que cumplan unas especificaciones determinadas.
— Ordenar y reajustar un conjunto de elementos de datos en una secuencia u orden determinado. (Ejemplo: en una aplicación de etiquetas de direcciones, el tener todos los nombres y direcciones ordenados por prefijo telefónico puede ser de gran ayuda.)
— Fundir diferentes bases de datos del mismo tipo para un fin específico.

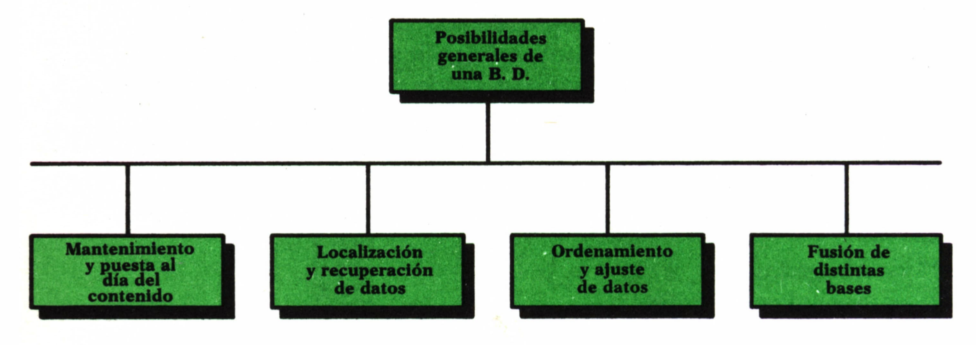


Fig. 2.1.

Desde el profesional de la informática hasta el empresario, investigador o profesional independiente, todos trabajamos con datos. Puesto que es un tema que nos incumbe a todos, veamos qué ventajas nos supone trabajar con una base de datos:

— Posibilidad de manipular bloques de información tan grandes como las bases de datos o tan pequeños como un carácter.
— Facilidad a la hora de añadir, insertar o borrar información de los ficheros, lo cual supone no sólo una mayor comodidad, sino también un ahorro de tiempo y dinero.
— Hallar la información necesaria en cuestión de segundos.
— Mantener en orden los ficheros.

— Realizar distintas tareas dependiendo del día de la semana, o de otro criterio que se escoja.

— Seguridad de la información

— Producción de informes automáticos.

— Combinación de ficheros según se requiera para tareas complejas de administración.

— Fácil generación de cartas: el hecho de enviar un número de cartas elevado, todas ellas personalizadas, sin una base de datos, supone un trabajo lento, costoso y pesado. Sin embargo, si las cartas se generan con la ayuda de una base de datos, esta tarea se simplifica enormemente.

Esquemáticamente las bases de datos permiten, por tanto:

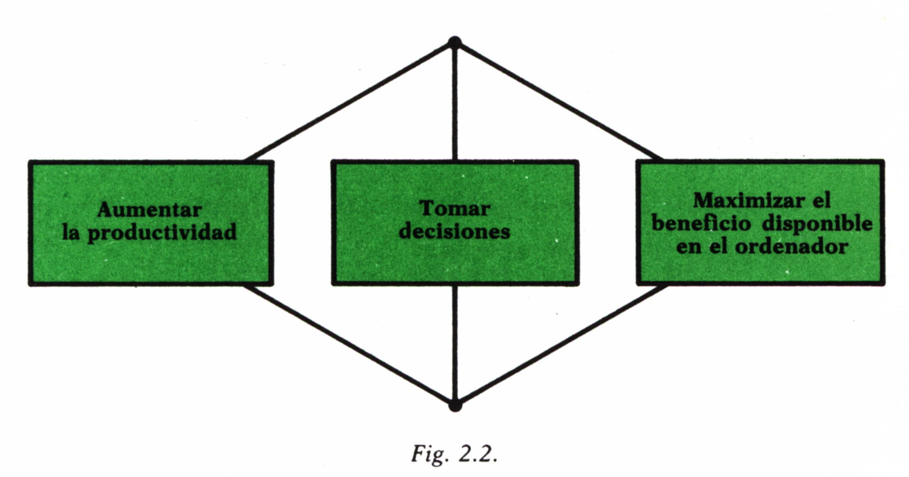


Fig. 2.2.

Dividiremos las aplicaciones prácticas de una base de datos según el sector al que nos estemos refiriendo según los siguientes tres bloques:

SECTOR GESTION DE EMPRESAS
SECTOR PROFESIONALES INDEPENDIENTES
OTROS SECTORES



SECTOR GESTION DE EMPRESAS



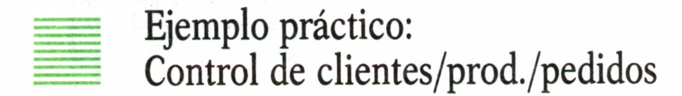
Operaciones típicas

Tanto para una empresa informatizada como para una empresa en vías de informatización, está claro que, en una gestión dinámica, es necesario tener un control directo, rápido y total de los datos. A cualquier empresa

le es útil contar con personal capaz de manejar directamente los datos. Y es precisamente el relativo fácil manejo de bases de datos el que permite que, sin tener conocimientos a nivel informático, las tareas de gestión de datos las puedan realizar desde los directivos hasta el personal encargado de la introducción de los datos.

Centrándonos un poco sobre el terreno práctico, citaremos algunos de los trabajos que se pueden ver simplificados mediante el manejo de una base de datos:

— FACTURACION
— CONTABILIDAD
— DECLARACIONES DE HACIENDA
— GESTION DE TRABAJOS DE SECRETARIA
— ELABORACION DE NOMINAS
— CONTROL DE CORRESPONDENCIA
— GENERACION DE CARTAS
— CONTROL DE CLIENTES



Ejemplo práctico: Control de clientes/prod./pedidos

En el ejemplo de control de clientes veamos qué posibilidades tendríamos con nuestra base de datos, si ésta fuese relacional. Supongamos que en la empresa tenemos un fichero de clientes, otro de productos y otro de pedidos.

Fichero de clientes

Los registros de este fichero constarán de los siguientes campos:

Nombre	Saldo	Actualización	Vencido
1	2	3	4

1) **Nombre:** Tipo alfanumérico (cadenas de letras, números y caracteres especiales).

2) **Saldo:** Tipo decimal (número positivo o negativo, con o sin coma decimal).

3) **Actualización:** Tipo: fecha (fecha de calendario)

Ej.: 10-1-85
9-3-86

4) **Vencido:** Tipo: verdad/falso

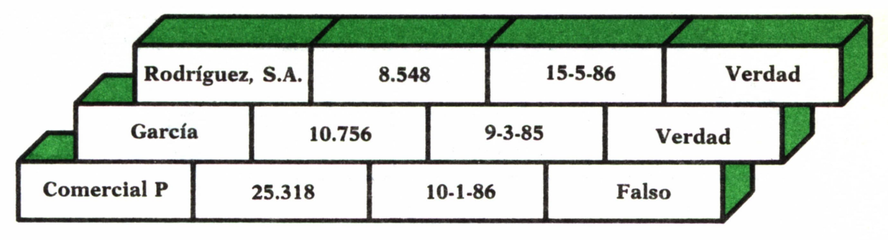


Fig. 2.3.

Fichero de pedidos

Los registros de este fichero constarán de los siguientes campos:

Nombre	Producto	Pedido	Cantidad	Total
1	2	3	4	5

1) Análogo al caso anterior.
2) **Producto:** Tipo: numérico. Observ.: A cada producto se le asigna un código numérico.
3) **Pedido:** Tipo: fecha. Observ.: Refleja la fecha en que se ha efectuado el pedido.
4) **Cantidad:** Tipo: numérico. Observ.: Indica la cantidad pedida de ese producto.
5) **Total:** Tipo: numérico.

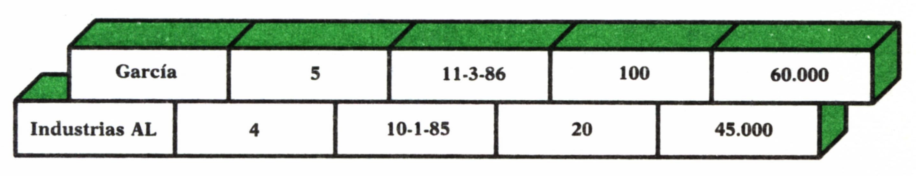


Fig. 2.4.

Fichero de productos

Consta de registros formado de los siguientes campos:

Producto	Descrip.	Precio
1	2	3

1) **Producto:** Análogo al caso anterior.
2) **Descrip.:** Tipo: alfanumérico. Observ.: Indica el nombre del producto asociado al código que haya en 1).
3) **Precio:** Tipo: Numérico.

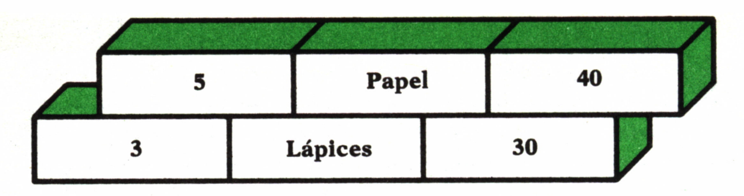


Fig. 2.5.

A continuación veamos algunas operaciones que podríamos realizar con estos ficheros, si dispusiéramos de una B.D. relacional, como puede ser dBASE III u OPEN ACCESS.

— Obtener un listado con todos los registros de todos los clientes con todos los campos: El listado sería:

Nombre	Saldo	Actualización	Vencido
Almacenes Pepe	8.544	10.1.83	Verdad
Azúcares, S. A.	9.655	11.2.85	Verdad
Blanco	10.844	12.3.86	Falso

Fig. 2.6.

— Obtener todos los registros del fichero de clientes, donde sólo aparezcan los campos nombre, saldo y vencido:

Nombre	Saldo	Vencido
Almacenes Pepe	8.544	10.1.85
Azúcares, S. A.	9.655	11.2.85
Blanco	10.844	12.3.86

Fig. 2.7.

— Recuperar los registros del fichero de clientes, cuyo saldo sea mayor que 30.000.

Nombre	Saldo	Actualización	Vencido
Breg, S. A.	35.845	16.1.85	Falso
Cras	41.865	21.3.86	Verdad
Darío, S. A.	67.844	22.2.85	Verdad

Fig. 2.8.

Obsérvese que los valores de saldo son todos mayores que 30.000.

— Del fichero de pedidos recuperar aquéllos cuya inicial del nombre sea mayor que M y número de producto menor que 50.

Nombre	Producto	Pedido	Cantidad	Total
Olea	35	15.2.85	41	8.615
Pasaman	47	27.3.86	24	15.000

Fig. 2.9.

— Del fichero de productos recuperar aquéllos cuyo número de producto sea mayor que 100 y cuyo precio sea menor que 10.000.

Producto	Descripción	Precio
110	Tijeras	245
144	Lámpara	7.545
235	Timbre	2.100
247	Silla	9.300

Fig. 2.10.

Otras operaciones posibles serían:

— Introducir los datos de un nuevo cliente/producto o pedido en su fichero correspondiente.

— Confrontar el valor de un campo de un fichero con los valores de otro campo en otro fichero para evitar introducir valores incorrectos.

— Recuperar los registros de clientes actualizados después de una fecha determinada.

— Clasificar el fichero de pedidos por número de producto.

— Modificar la información de los registros de uno de los ficheros, uno por uno.

— Borrar un registro del fichero.

— Generar un informe con todos los pedidos realizados por la empresa en un día determinado.

— Escribir mensualmente una carta a los clientes con saldos pendientes.

— Ampliar el tamaño de uno de los ficheros porque se ha quedado corto.

— Añadir un nuevo campo a un registro de uno de los ficheros que originalmente no existía. Por ejemplo, en el fichero de productos podríamos añadir un nuevo campo para la marca del producto.

Etc.

A la vista de lo expuesto, podemos hacernos una idea de las aplicaciones que se podrían realizar, la facturación, contabilidad, etc.



SECTOR PROFESIONALES INDEPENDIENTES

Introducción

Gran parte de usuarios que utilizan ordenadores personales son profesionales de profesiones liberales. Pueden ser:

— Gestores de datos de pequeñas empresas.
— Publicistas.
— Médicos.
— Abogados.
— Matemáticos.
— Ejecutivos.
— Biólogos.
— Geólogos.
— Arquitectos, etc.

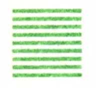
Sus profesiones les obligan a ser muy elásticos y a tener en todo momento control personal y eficaz de los datos que emplean. Por ello, no pueden depender del mercado del software a medida, ya que necesitan las utilidades casi en el momento en que las diseñan.

Necesitan multitud de programas que puedan ser alterados a diario. Dado que, generalmente, no pueden adquirir paquetes a medida, porque no suelen existir, o porque su coste resultaría elevado para la utilidad que le van a dar, normalmente adquieren un programa estándar que se dedica a su rama, pero necesitan manipular otros datos o procesos no incluidos.



Ejemplos

Veamos las utilidades que podrían sacar algunos de estos profesionales con ayuda de su base de datos.



Médicos

Para un médico el hecho de disponer de una base de datos puede ser de suma utilidad. Con ella podría tener un fichero de pacientes ordenado, bien alfabéticamente o bien por cualquier otro orden que le conviniera.

En él podría reflejar la información requerida para cada paciente: nombre, dirección, teléfono, edad, sociedad a la que está afiliado (caso de estarlo), número de afiliación, e incluso un breve historial clínico esquema-

tizado. De esta forma, a la hora de atender a un paciente, la búsqueda de su «ficha» personal resultará mucho más rápida y fácil que si esta información se recogiera manualmente.

Le permitiría también llevar un control de aquellos pacientes que tuvieran que ser citados para consulta, o bien pacientes a ser ingresados para operar, etc.

Si nos adentramos en el caso de un hospital, se observa que las ventajas serían aún mayores. Dado que tratan con un número elevado de pacientes, el llevar un control más estricto y riguroso sobre la información concerniente a cada uno de ellos resultará no sólo práctico, sino más bien necesario. Si a la información citada anteriormente como: nombre, dirección, teléfono, edad, sociedad a la que está afiliado, número de afiliación e historial clínico le añadimos: fecha de alta en el hospital, fecha de baja, médico por el que ha sido tratado, período de recuperación, resultados, análisis efectuados, etc., se podrían obtener con facilidad informes sobre:

— Relación de enfermos a dar de alta en un día determinado.

— Relación de pacientes a llamar para su ingreso en el hospital.

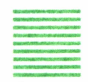
— Relación de enfermos a pasar consulta en un día o período determinado.

— Informe de pacientes pendientes de pago.

— Relación de pacientes a dar de baja en un día o plazo determinado.

— Etc.

Es evidente que con este control se evitarían muchos problemas que existen actualmente por falta de planificación y organización de los hospitales.



Abogados

No hace falta extendernos mucho para darnos cuenta de la utilidad que puede suponer para un abogado el disponer de una base de datos. Para cada cliente podría guardar, por ejemplo: nombre, dirección, teléfono, tipo de cliente (nuevo/antiguo), casos que se han tratado con él, tipo de caso (laboral, matrimonial...) y toda la información adicional que se requiera. De esta forma, diaria, semanal o mensualmente podría obtener un listado de clientes pendientes de efectuar determinados trámites, determinar qué trámites están pendientes para un cliente en particular, recuperar información disponible de un cliente, listar los clientes pendientes de cobro, etcétera.



Biólogos

Supóngase un biólogo que disponga de un laboratorio particular y que desee, por ejemplo, elaborar un herbario recogiendo la información recopilada para un conjunto de plantas. Sería interesante que en la base de datos guardase los datos de cada una de ellas, como puede ser: código, nombre, nombre científico, familia, procedencia, características específicas, enfermedades que puede sufrir, etc. Así podría en un momento dado recuperar un dato concreto de uno de los ejemplares, o bien elaborar un informe con todas las plantas de una misma familia, o bien añadir los datos de una nueva planta, etc.



OTROS SECTORES/APLICACIONES VARIAS

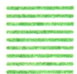


Introducción

En esta sección incluiremos otras aplicaciones de las bases de datos de carácter menos general que las citadas anteriormente o bien de menor complejidad. No pretendemos dar aquí una selección rigurosa y ordenada de las aplicaciones, sino más bien algunas de las restantes aplicaciones que permiten las bases de datos.

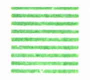


Ejemplos



Colecciones

Mediante las bases de datos podemos manejar también la información que tengamos de una determinada colección (ya sea de sellos, monedas...). Si para cada ejemplar de la colección recogemos: nombre, año, país, colección a la que pertenece, procedencia..., en un momento dado podremos sacar un listado con todos los ejemplares que existen (caso de tenerlos registrados en el ordenador) o bien obtener un listado por orden alfabético de todos los ejemplares, o también recuperar toda la información existente referida a una única pieza, etc. Este tipo de aplicación no requiere, además, una base de datos demasiado potente para su realización, así que cualquier aficionado a la informática con un micro pequeño y una base de datos sencilla de manejo podría disfrutar de ella.



Referencias bibliográficas

Todos hemos necesitado alguna vez encontrar un artículo determinado de una revista o periódico atrasado y hemos hallado dificultades para buscarlo. Podríamos crear una base de datos que contuviera para cada revista o periódico que nos interese:

— Relación de artículos aparecidos en un ejemplar.
— Fecha de publicación.
— Distribuidor de la publicación.
— Extracto de cada artículo.

Así, podríamos localizar todos los artículos aparecidos en un período fijo de tiempo referentes a un mismo tema, o bien encontrar un artículo determinado (en qué ejemplar apareció) o bien ordenar todas las publicaciones según los distribuidores, etc.



Calendario de citas

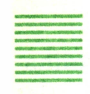
La elaboración de un calendario de citas mediante una base de datos también es posible. Podríamos tener un calendario actualizado para la semana, mes o año próximo tal que, al concertar una nueva cita, consultásemos y comprobásemos si estamos intentando hacer una planificación contradictoria. Si el calendario lo elaboramos manualmente, es probable que, para una misma fecha y hora, tratemos de concertar dos citas distintas. Con la base de datos este problema no tendría por qué existir. Podríamos, además, obtener una relación de todas las citas dentro de un lapso de tiempo determinado (la próxima semana, el mes que viene...), o bien listar todas aquellas citas a anular por existir problemas.



Alquiler

Si nos encontramos en el caso de un centro de alquiler como puede ser, por ejemplo, un vídeo-club, el disponer de una base de datos también resultará de gran utilidad. En la base de datos recogeremos todas las películas de que se dispone, a qué cliente se han alquilado, fecha de alquiler, fecha a devolver, precio de la película, etc., de tal forma que, por ejemplo, a diario se podría obtener una relación de películas a devolver con los números de teléfono de los clientes que las tienen, ordenar las películas por temas (oeste, aventuras, ciencia-ficción, policíacas) y sacar un listado cuan-

do interese de qué películas hay disponibles para uno de los sistemas (Beta, VHS, 2000) dentro de un tema concreto...



Club social

En un club social en el cual se maneja un archivo de socios de gran tamaño una base de datos concreta sería rentable. Recogeríamos en ellos los datos de todos los socios, desde nombre, dirección, teléfono, edad, deportes practicados, fecha de inscripción, cuota mensual-anual a pagar, número de personas que disfruta de un mismo número de afiliación, etc.

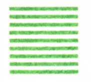
Las operaciones que hubiera que realizar diaria, semanal, mensual o anualmente como: listado de clientes a pagar cuota atrasada o reservas hechas a una hora determinada para unas instalaciones o listado de nuevos socios inscritos en un plazo determinado, o altas de nuevos socios, etc., con el uso de una base de datos sería relativamente sencillo.



Comercios

En cualquier comercio, grande o pequeño, donde se vendan numerosos productos, sean del tipo que sean, el manejo de éstos mediante una base de datos resulta altamente beneficioso.

Recogeríamos en ella los datos de cada producto como puedan ser: nombre, tipo de producto, precio unitario, distribuidor, stock existente, stock óptimo, número de productos vendidos mensualmente, fecha del último pedido, etc. En cualquier momento podríamos sacar un listado de todos aquellos productos cuyo stock estuviera por debajo del óptimo o bien una relación de todos los productos distribuidos por una misma empresa, sacar un listado de aquellos productos más vendidos mensual-anualmente, o sacar un listado de aquellos menos vendidos para disminuir la compra de los mismos, etc.



Bibliotecas

Una biblioteca es un ejemplo típico en el cual el uso de una base de datos documental es extremadamente útil. Las tareas que se pueden automatizar con una base de datos son:

a) CONTROL DE ADQUISICIONES.
b) CATALOGACION.
c) CONTROL DE PRESTAMOS.
d) CONTROL DE PUBLICACIONES PERIODICAS.

Nos extenderemos más en la explicación de esta aplicación por ser de sobra conocida por la mayoría de nosotros:

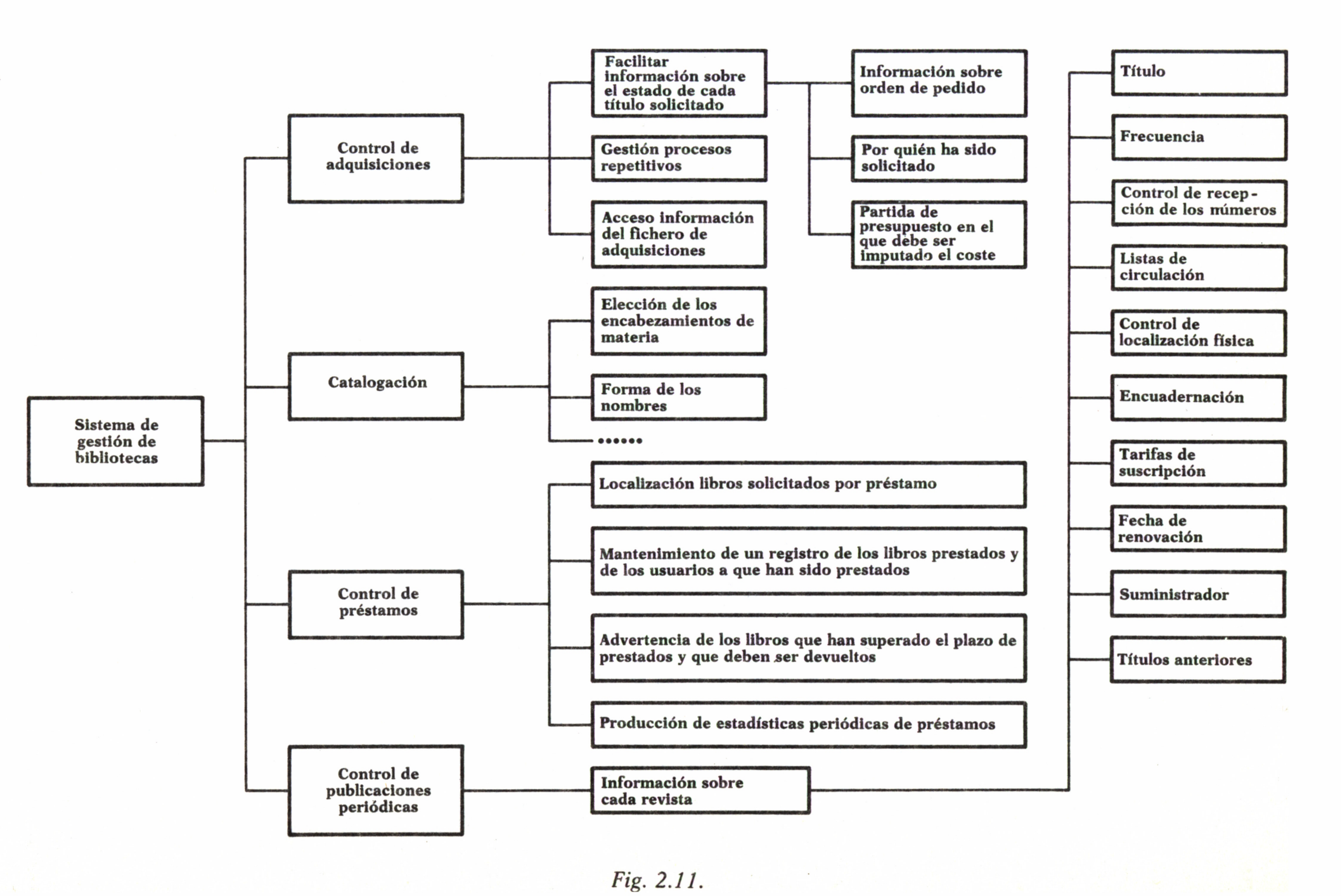

Sistema de gestión de bibliotecas
Control de adquisiciones
Facilitar información sobre el estado de cada título solicitado
Información sobre orden de pedido
Por quién ha sido solicitado
Partida de presupuesto en el que debe ser imputado el coste
Gestión procesos repetitivos
Acceso información del fichero de adquisiciones
Catalogación
Elección de los encabezamientos de materia
Forma de los nombres
••••••
Control de préstamos
Localización libros solicitados por préstamo
Mantenimiento de un registro de los libros prestados y de los usuarios a que han sido prestados
Advertencia de los libros que han superado el plazo de prestados y que deben ser devueltos
Producción de estadísticas periódicas de préstamos
Control de publicaciones periódicas
Información sobre cada revista
Título
Frecuencia
Control de recep-ción de los números
Listas de circulación
Control de localización física
Encuadernación
Tarifas de suscripción
Fecha de renovación
Suministrador
Títulos anteriores

Fig. 2.11.

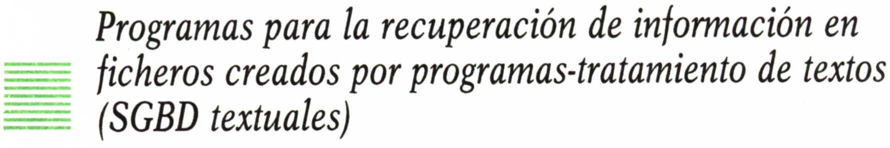
a) Control de adquisiciones. El propósito de un fichero de adquisiciones es facilitar información sobre el estado de cada título solicitado por la biblioteca, es decir, información sobre la orden de pedido, (suministrado, fecha, etc.) por quién ha sido solicitado y la partida de presupuesto en el que debe ser imputado su coste. También puede advertir sobre los títulos que se han solicitado y no se han recibido, indicando las razones del retraso y contestar a las preguntas de los usuarios que desean averiguar el progreso de sus solicitudes.

El control de adquisiciones es a menudo un proceso mecánico y repetitivo en el que es frecuente que una misma información se repita en varias etapas del proceso de petición al suministrador (los mismos datos de autor, título, editor, etc., deben aparecer en ficheros de registro, formularios de pedido, listas de títulos no recibidos, etc.). El microordenador es ideal para gestionar estos procedimientos, permitiendo además establecer muchos más puntos de acceso a la información contenida en el fichero de adquisiciones.

b) Catalogación. Todavía sigue siendo necesario el esfuerzo intelectual del catalogador: aunque algunos programas incluyen automáticamente la puntuación y los signos requeridos por las diversas normas de catalogación, formatean las entradas, etc., las decisiones sobre la elección de los encabezamientos de materia, la forma de los nombres, etc., recaen en el bibliotecario, que se basa para adoptarlas en el conocimiento de normas específicas de catalogación y en su práctica profesional.

c) Control de préstamos. Para asegurar el control de préstamos es preciso: localizar los libros solicitados para préstamo, mantener un registro de los libros prestados y de los usuarios a los que han sido prestados, advertir sobre los libros que han superado el plazo de préstamo y deben ser renovados o devueltos. Podrían incluir también la producción de estadísticas periódicas de préstamos.

d) Control de publicaciones periódicas. El fichero necesario para hacerlo posible debería contener una información reducida sobre cada revista: título, frecuencia, control de recepción de los números, listas de circulación, control de localización física, encuadernación, tarifas de suscripción y fecha de renovación, suministrador y títulos anteriores. En el caso de las hemerotecas no son necesarias todas las capacidades y potencia de los Sistemas de Gestión Integrada de Bibliotecas.



Programas para la recuperación de información en ficheros creados por programas-tratamiento de textos (SGBD textuales)

Un problema corriente en las oficinas que utilizan intensivamente el tratamiento de textos para crear documentos es el de localizarlos más tar-

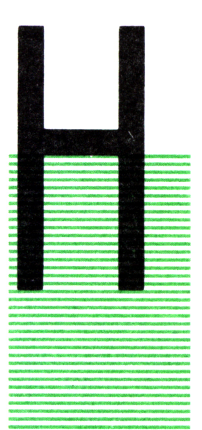
de entre centenares de ficheros contando tan solo como pista con los caracteres que permite el PC para identificar el fichero.

Para resolver este problema, los S.G.B.D. catalogan y organizan los ficheros de documentos basándose en las palabras o frases que éstos contienen. Estos programas exploran los ficheros de texto, palabra por palabra, y crean una base de datos que las ordenan mediante un índice. Las palabras vacías de sentido (artículos, etc.) son eliminadas durante este proceso y sólo se retienen nombres, adjetivos, etc., para crear el índice. Una vez creada la base de datos, el usuario puede localizar los documentos deseados interrogándola con el término o las combinaciones de términos que describen esos documentos.

Algunos gestores de este tipo nos permiten también:

— Realizar búsquedas de frases o grupos de palabras.
— Combinar palabras usando lógica Booleana (and, or, not).
— Efectuar búsquedas contextuales que hacen posible la recuperación de documentos que contienen ciertas palabras o frases próximas entre sí.
— Ojear un documento una vez localizado.
— Describir cada documento con un breve comentario que permite confirmar si se ha localizado el documento deseado.

EJEMPLO PRACTICO 3



HASTA ahora hemos visto una serie de conceptos teóricos de lo que es una base de datos y qué posibles aplicaciones tiene. Ahora vamos a realizar un ejemplo práctico para ver el funcionamiento real de una base de datos concreta.

La base de datos escogida es la dBASE III, por ser una de las más conocidas y utilizadas en estos momentos. Más adelante veremos con detalle las especificaciones técnicas y las posibilidades del dBASE III; por ahora basta con decir que es una de las bases de datos de uso general más potentes que existen en el mercado actualmente.

Durante todo el desarrollo del ejemplo vamos a utilizar una función del dBASE III llamada ASSIST, que permite trabajar de una forma sencilla, a base de menús, para que se vea de una forma más clara su utilización y funcionamiento.

El ejemplo va a consistir en el desarrollo de una simple agenda telefónica en la que incluirán los siguientes datos:

— NOMBRE

— APELLIDOS

— DIRECCION

— CIUDAD

— N. DE TELEFONO

— FECHA DE NACIMIENTO

Con todos estos datos formaremos una base de datos que luego utilizaremos para buscar datos relativos a una persona determinada o a partir de

preguntas concretas obtener cuál es o son las personas que satisfacen esa pregunta, como, por ejemplo: «Es hoy el cumpleaños de alguien».



CREACION DE LA AGENDA

Lo primero que tenemos que hacer antes de usar la base de datos es crear su estructura.

Para ello usamos la función CREATE. Esta nos permite definir el modelo de un registro para definir cuáles van a ser los campos, su nombre y significado.

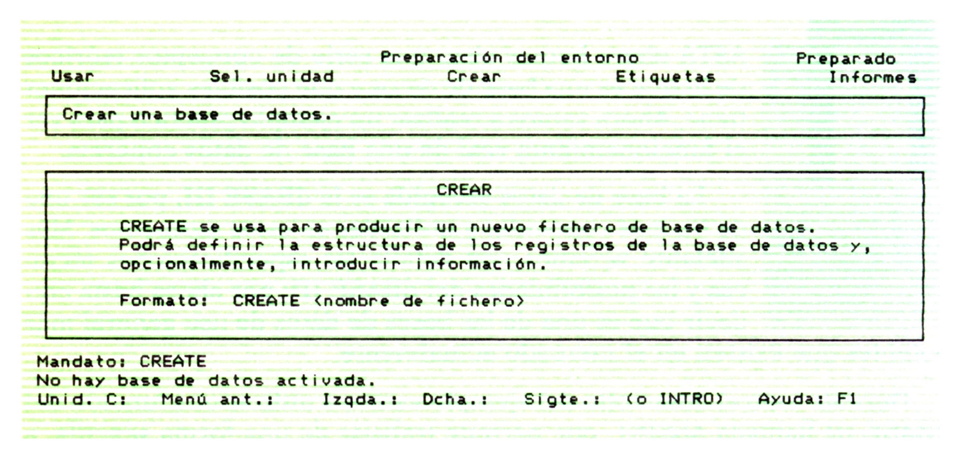


Fig. 3.1.

El fichero de la base de datos debe tener un nombre y esto es lo primero que hemos de hacer.

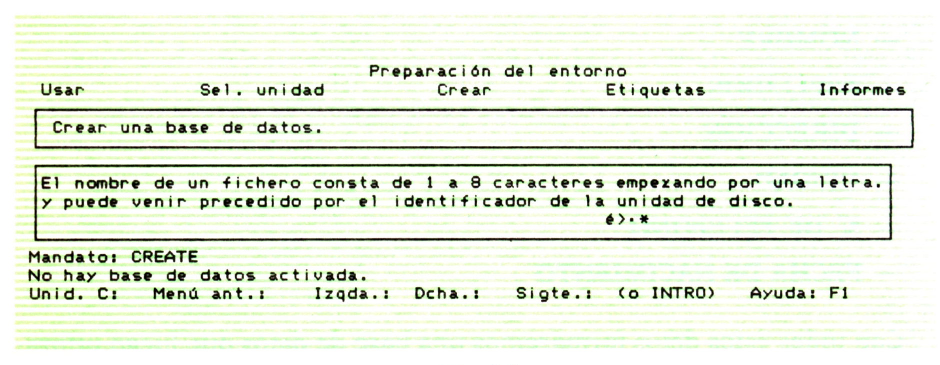


Fig. 3.2.

El nombre que le vamos a dar a nuestra base de datos es, evidentemente, «agenda»

Ahora tenemos que definir cada uno de los campos de un registro de la base de datos.

```
C:agenda.dbf                                          Bytes restantes:   3930
                                                      Campos definidos:     6
   Nomb campo  Tipo       Ancho  Dec               Nomb campo  Tipo      Ancho  Dec

 1 NOMBRE      Car/texto   10
 2 APELLIDOS   Car/texto   20
 3 DIRECCION   Car/texto   15
 4 CIUDAD      Car/texto   10
 5 TELEFONO    Numérico     7    0
 6 FECHA_NAC   Día/Fecha    8

Campos de Caracteres pueden contener todo tipo de caracteres alfanuméricos
```

Fig. 3.3.

Como podemos ver en la figura 3.3, tenemos definidos 6 campos, cada uno corresponde a la información que queremos guardar de cada persona. En la definición de cada campo aparecen una serie de datos por el siguiente orden:

— **Nombre campo:** Es el nombre por el que vamos a identificar el campo.

— **Tipo:** Es el tipo de datos que se van a guardar en el campo, como, por ejemplo, caracteres, números o fechas.

— **Ancho:** Es la longitud de cada campo. En el caso de que sea un campo de caracteres, es el número de letras que caben. Si el campo es numérico, es el número máximo de cifras que va a tener ese campo, etc.

— **Dec.:** En el caso de que el campo sea numérico en esta posición introducimos el número de decimales que va a tener el número.

Y con estos datos definidos para cada uno de los campos ya tenemos definida nuestra base de datos.

Ahora hay que introducir los datos en la base. Esto lo hacemos con un comando llamado APPEND (añadir), que nos permite ir añadiendo datos a la base llenando todos o algunos de los campos de cada registro.

En la figura 3.4 tenemos un ejemplo de introducción de datos en dos registros distintos. Una vez lleno el primero, pasamos a rellenar el siguiente, y así hasta que tengamos todos los datos introducidos.

```
Reg. Nº        1
NOMBRE         JUAN
APELLIDOS      PEREZ GONZALEZ
DIRECCION      FUENTE NUEVA 10
CIUDAD         MADRID
TELEFONO       4490033
FECHA_NAC      20-03-55

Reg. Nº        2
NOMBRE         JOSE MARIA
APELLIDOS      ZAPATERO GARCIA
DIRECCION      CONDE AGRAZ 78
CIUDAD         MADRID
TELEFONO       4425690
FECHA_NAC      06-10-61
```

Fig. 3.4.

Una vez introducidos todos los datos podemos verlos todos en forma de tabla con un comando que nos permite recorrer la base de datos registro a registro y movernos entre los campos. Esta función se llama BROWSE (otear). En la figura 3.5 tenemos un ejemplo de su funcionamiento.

```
Reg. Nº            1    agenda
NOMBRE---- APELLIDOS----------- DIRECCION------ CIUDAD---- TELEFONO FECHA_NAC
JUAN       PEREZ GONZALEZ       FUENTE NUEVA 10 MADRID      4490033 20-03-55
JOSE MARIA ZAPATERO GARCIA      CONDE AGRAZ 78  MADRID      4425690 06-10-61
ANA MARIA  PEREZ GONZALEZ       FUENTE NUEVA 10 MADRID      4490033 12-07-57
MIGUEL     BARROSO PEREZ        PIMIENTA 30     SEVILLA      332456 09-05-63
ANGEL      MARTINEZ GOMEZ       VILLANUEVA 77   MADRID      6576484 03-12-60
LUISA      PEREZ  LOPEZ         ALCALA  260     MADRID      4037846 14-10-60
FERNANDO   ALBARRAN GONZALEZ    DOCTOR MUR  3   SEGOVIA       78564 04-04-50
```

Fig. 3.5.



RECUPERACION DE INFORMACION

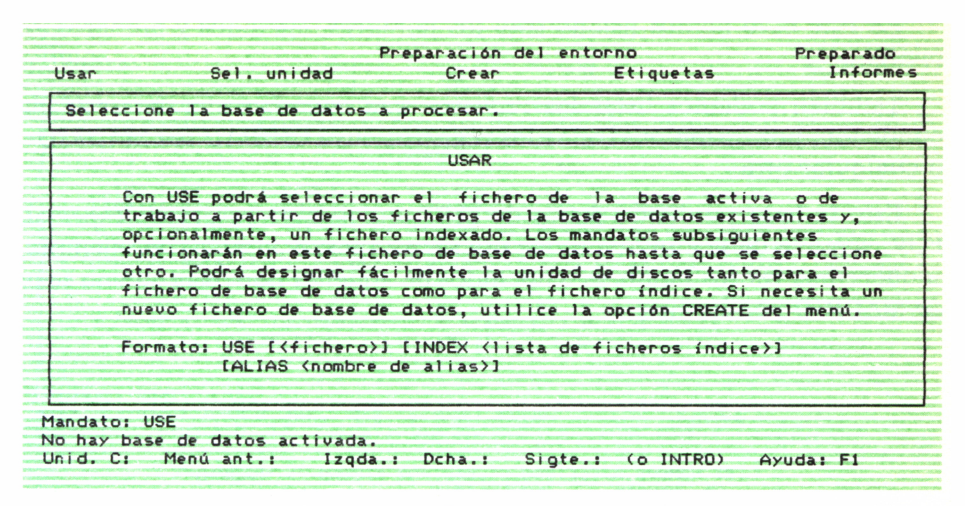
Una vez creada la agenda, ya podemos utilizarla para buscar datos de alguna persona.

Lo primero que tenemos que decirle a la base de datos es el fichero que queremos emplear. Esto se hace mediante el mandato USE (utilizar). Con este mandato le decimos cuál va a ser el fichero en el que vamos a buscar datos. En este caso el mandato sería USE AGENDA.

Ahora estamos listos para recuperar datos o añadir nuevos registros.

Para ver algún dato en concreto se usa la instrucción LIST, que lo que hace es listar los registros de la base de datos.

Si introducimos este comando solo, sin ningún parámetro, entonces nos muestra toda la base de datos, en este caso toda la agenda telefónica.



```
                     Preparación del entorno                    Preparado
Usar            Sel. unidad           Crear          Etiquetas          Informes

 Seleccione la base de datos a procesar.

                                    USAR

    Con USE podrá seleccionar el  fichero de  la  base  activa  o de
    trabajo a partir de los ficheros de la base de datos existentes y,
    opcionalmente, un fichero indexado. Los mandatos subsiguientes
    funcionarán en este fichero de base de datos hasta que se seleccione
    otro. Podrá designar fácilmente la unidad de discos tanto para el
    fichero de base de datos como para el fichero índice. Si necesita un
    nuevo fichero de base de datos, utilice la opción CREATE del menú.

    Formato: USE [<fichero>] [INDEX <lista de ficheros índice>]
                 [ALIAS <nombre de alias>]

Mandato: USE
No hay base de datos activada.
Unid. C:  Menú ant.:   Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

Fig. 3.6.

```
. list
Reg. Nº  NOMBRE      APELLIDOS           DIRECCION          CIUDAD     TELEFONO FEC
HA_NAC
      1  JUAN        PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID        4490033 20-
03-55
      2  JOSE MARIA ZAPATERO GARCIA     CONDE AGRAZ 78    MADRID        4425690 06-
10-61
      3  ANA MARIA   PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID        4490033 12-
07-57
      4  MIGUEL      BARROSO PEREZ       PIMIENTA 30        SEVILLA      332456 09-
05-63
      5  ANGEL       MARTINEZ GOMEZ      VILLANUEVA 77      MADRID      6576484 03-
12-60
      6  LUISA       PEREZ  LOPEZ        ALCALA  260        MADRID      4037846 14-
10-60
      7  FERNANDO    ALBARRAN GONZALEZ   DOCTOR MUR  3      SEGOVIA       78564 04-
04-50
```

Fig. 3.7.

Sin embargo, podemos hacer que sólo nos muestre uno o varios campos determinados de la base de datos. Esto se hace poniendo detrás de la palabra LIST los nombres de los campos que queremos ver. Por ejemplo:

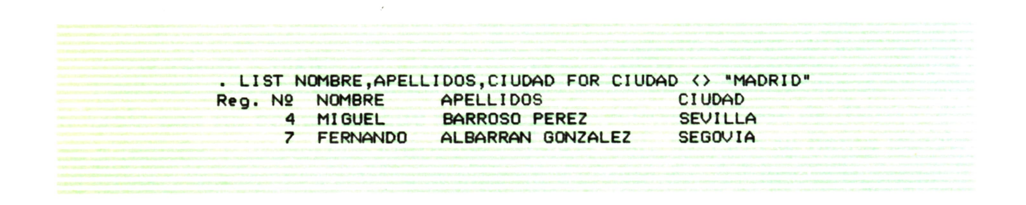
```
. list nombre,apellidos,telefono
Reg. Nº  nombre      apellidos            telefono
      1  JUAN        PEREZ GONZALEZ        4490033
      2  JOSE MARIA  ZAPATERO GARCIA       4425690
      3  ANA MARIA   PEREZ GONZALEZ        4490033
      4  MIGUEL      BARROSO PEREZ          332456
      5  ANGEL       MARTINEZ GOMEZ        6576484
      6  LUISA       PEREZ  LOPEZ          4037846
      7  FERNANDO    ALBARRAN GONZALEZ       78564
```

Fig. 3.8.

En este caso hemos listado sólo los nombres, apellidos y números de teléfono.

También podemos listar los registros que cumplan una determinada condición, por ejemplo: «Todos los registros cuya CIUDAD sea MADRID», o cualquier otra.

Esto se hace poniendo detrás de LIST una palabra reservada especial, FOR, seguida de la condición que queremos que se cumpla. Por ejemplo, si queremos ver los nombres o apellidos de aquellas personas que no sean de MADRID pondremos:



```
. LIST NOMBRE,APELLIDOS,CIUDAD FOR CIUDAD <> "MADRID"
Reg. Nº  NOMBRE      APELLIDOS             CIUDAD
      4  MIGUEL      BARROSO PEREZ         SEVILLA
      7  FERNANDO    ALBARRAN GONZALEZ     SEGOVIA
```

Fig. 3.9.

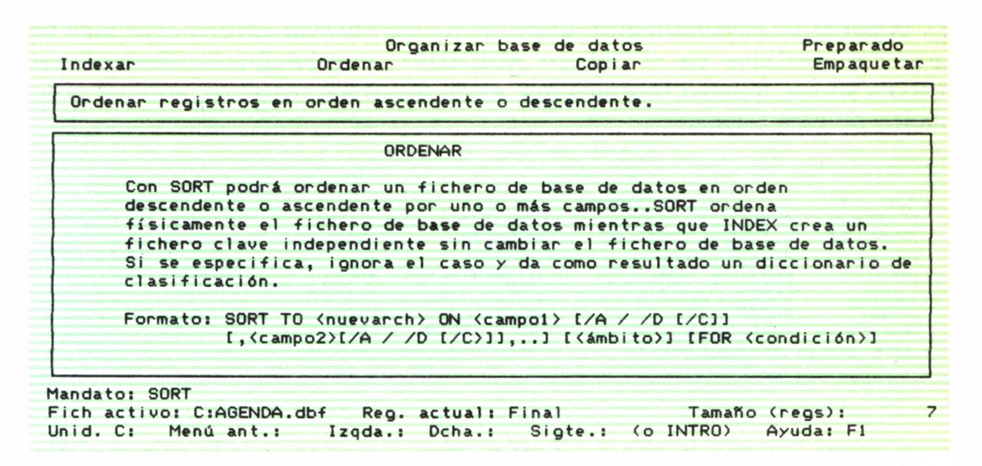
Como vemos, sólo se muestran en este caso dos registros, aquellos que no tienen en el campo CIUDAD la palabra MADRID.



ORDENACION

Hasta ahora hemos introducido los datos en la agenda según nos van llegando, pero no guardan entre sí ningún orden especial. Si tuviéramos que introducir los datos según un orden determinado, por ejemplo, ordenado por apellidos, entonces tendríamos el problema de que si más tarde tenemos que introducir otro dato cuyo orden fuera anterior al último dato metido, deberíamos hacer un hueco en el fichero en el lugar correspondiente para poder dejar el fichero ordenado. En realidad, es más cómodo introducir los datos según vengan y una vez finalizado el proceso de introducción de datos ordenar nuestra base de datos.

Para esto disponemos de un mandato SORT (ordenar), que nos permite precisamente eso, ordenar una base de datos según un campo o campos determinado produciendo otro fichero en el que los datos ya están ordenados.



```
                              Organizar base de datos                Preparado
  Indexar                 Ordenar                 Copiar              Empaquetar

 Ordenar registros en orden ascendente o descendente.

                              ORDENAR

      Con SORT podrá ordenar un fichero de base de datos en orden
      descendente o ascendente por uno o más campos..SORT ordena
      físicamente el fichero de base de datos mientras que INDEX crea un
      fichero clave independiente sin cambiar el fichero de base de datos.
      Si se especifica, ignora el caso y da como resultado un diccionario de
      clasificación.

      Formato: SORT TO <nuevarch> ON <campo1> [/A / /D [/C]]
               [,<campo2>[/A / /D [/C>]],..] [<ámbito>] [FOR <condición>]

 Mandato: SORT
 Fich activo: C:AGENDA.dbf   Reg. actual: Final          Tamaño (regs):       7
 Unid. C:   Menú ant.:    Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

Fig. 3.10.

Para ordenar un fichero lo primero que tenemos que hacer es elegir el campo o campos por los cuales va a quedar ordenado, en este caso **APELLIDOS**.

```
                              Organizar base de datos
  Indexar                 Ordenar                 Copiar              Empaquetar

  Elegir campos. Sitúese con las teclas /. Active con INTRO

                              Nomb. campo    Tipo campo     Ancho    Nº Dec
                              NOMBRE          Car/texto      10
                             APELLIDOS       Car/texto      20
                              DIRECCION       Car/texto      15
                              CIUDAD          Car/texto      10
                              TELEFONO        Numérico        7        0
                              FECHA_NAC       Día/Fecha       8
 Mandato: SORT ON APELLIDOS
 Fich activo: C:AGENDA.dbf   Reg. actual: Final          Tamaño (regs):       7
 Unid. C:   Pulse  para pasar al siguiente elemento de selección.
```

Fig. 3.11.

Ahora debemos elegir un nombre para el fichero de salida, esto es, el fichero en el cual se va a copiar la base de datos, pero con los registros ordenados por apellidos.

```
                              Organizar base de datos
  Indexar                 Ordenar                 Copiar              Empaquetar

  Seleccione, si así lo desea, una condición

  100% ordenados             7 Registros ordenados
 Mandato: SORT ON APELLIDOS  TO AGENDA1
 Fich activo: C:AGENDA.dbf   Reg. actual: Final          Tamaño (regs):       7
 Unid. C:   Pulse cualquier tecla para continuar
```

Fig. 3.12.

A partir de este momento disponemos de nuestra agenda ordenada según los apellidos en el fichero **AGENDA1**, por lo que si ojeamos con **BROWSE** ese fichero veremos que están todos los registros, aunque ordenados de otra forma.

```
Reg. Nº             1    AGENDA1
NOMBRE---- APELLIDOS---------- DIRECCION------ CIUDAD---- TELEFONO FECHA_NAC
FERNANDO   ALBARRAN GONZALEZ   DOCTOR MUR  3   SEGOVIA       78564 04-04-50
MIGUEL     BARROSO PEREZ       PIMIENTA 30     SEVILLA      332456 09-05-63
ANGEL      MARTINEZ GOMEZ      VILLANUEVA 77   MADRID      6576484 03-12-60
LUISA      PEREZ  LOPEZ        ALCALA  260     MADRID      4037846 14-10-60
JUAN       PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID      4490033 20-03-55
ANA MARIA  PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID      4490033 12-07-57
JOSE MARIA ZAPATERO GARCIA     CONDE AGRAZ 78  MADRID      4425690 06-10-61
```

Fig. 3.13.

Esta forma de órdenes tiene el inconveniente de que, cada vez que introducimos un nuevo registro a nuestra agenda, tenemos que repetir todo el proceso de ordenación, lo cual resulta engorroso y molesto. Además, con esta técnica el espacio de almacenamiento en disco se reduce, ya que tenemos el mismo fichero duplicado en el disco.

Sin embargo, existe otra técnica, llamada indexación, que nos permite ordenar un fichero sin necesidad de crear un nuevo fichero de datos.

Esta técnica se basa en lo siguiente: la creación de un nuevo fichero, llamado fichero índice, en el cual sólo están almacenados los números de registro en el orden en que deberían estar ordenados según el mandato SORT.

```
                         Organizar base de datos               Preparado
Indexar                Ordenar               Copiar              Empaquetar

 Crear un fichero índice para la base de datos en uso.

                               INDEXAR

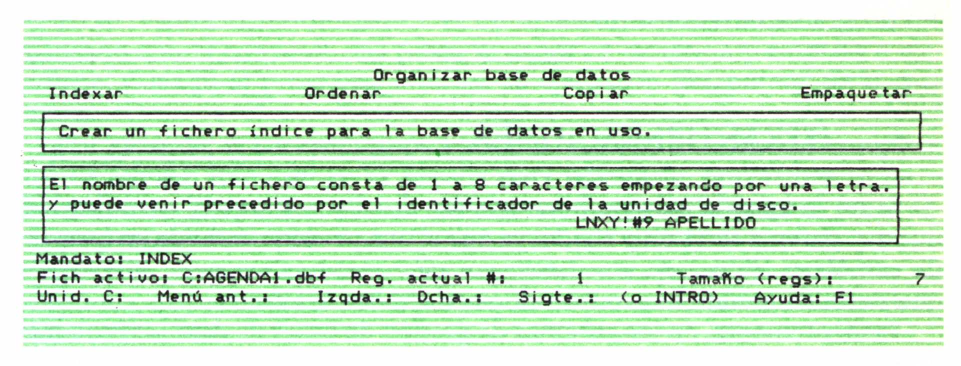
     INDEX se usa para crear un fichero índice para usarlo con el fichero
     de base de datos activo. Los ficheros índice facilitan la búsqueda
     rápida de información y permiten ver el contenido de la base de datos
     de forma ordenada. Hay una expresión clave que le indica a dBASE III
     el orden en el que desea buscar y ver los datos. Por ejemplo, una
     expresión clave podría indicarle a dBASE III que mantenga un fichero
     índice ordenado por el código postal.

     Formato: INDEX ON <expresión clave> TO <nombarch>

Mandato: INDEX
Fich activo: C:AGENDA1.dbf  Reg. actual #:     1      Tamaño (regs):       7
Unid. C:   Menú ant.:    Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

Fig. 3.14.

Para crear un archivo de índice, lo primero que tenemos que hacer es dar el nombre del fichero.



```
                              Organizar base de datos
Indexar                 Ordenar                 Copiar                 Empaquetar

 Crear un fichero índice para la base de datos en uso.

 El nombre de un fichero consta de 1 a 8 caracteres empezando por una letra,
 y puede venir precedido por el identificador de la unidad de disco.
                                        LNXY!#9 APELLIDO

Mandato: INDEX
Fich activo: C:AGENDA1.dbf  Reg. actual #:      1        Tamaño (regs):        7
Unid. C:   Menú ant.:    Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

Fig. 3.15.

Luego hay que elegir por qué campo se va a indexar el fichero y después realizar la indexación propiamente dicha.

```
                              Organizar base de datos
Indexar                 Ordenar                 Copiar                 Empaquetar

 Crear un fichero índice para la base de datos en uso.

      7 registros indexado(s)
Mandato: INDEX TO APELLIDO  ON APELLIDOS
Fich activo: C:AGENDA1.dbf  Reg. actual #:      1        Tamaño (regs):        7
Unid. C:   Pulse cualquier tecla para continuar
```

Fig. 3.16.

Si realizamos un **BROWSE** o un **LIST** al fichero original (**AGENDA**), veremos que aparece ordenado, pero en realidad el orden físico de los registros no ha cambiado, sino que cada vez que realizemos una operación el programa busca primero la tabla de índices que está activa en ese momento, siendo ese el orden que utiliza para realizarlas.

```
Reg. Nº          7    AGENDA
NOMBRE---- APELLIDOS---------- DIRECCION------ CIUDAD---- TELEFONO FECHA_NAC
FERNANDO   ALBARRAN GONZALEZ   DOCTOR MUR  3   SEGOVIA       78564 04-04-50
MIGUEL     BARROSO PEREZ       PIMIENTA 30     SEVILLA      332456 09-05-63
ANGEL      MARTINEZ GOMEZ      VILLANUEVA 77   MADRID      6576484 03-12-60
LUISA      PEREZ  LOPEZ        ALCALA  260     MADRID      4037846 14-10-60
JUAN       PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID      4490033 20-03-55
ANA MARIA  PEREZ GONZALEZ      FUENTE NUEVA 10 MADRID      4490033 12-07-57
JOSE MARIA ZAPATERO GARCIA     CONDE AGRAZ 78  MADRID      4425690 06-10-61
```

Fig. 3.17.

En este caso, si añadimos un nuevo registro no necesitaremos crear de nuevo el fichero de índices, ya que si está activo en ese momento el propio programa lo reconstruirá.



OTRAS OPERACIONES

Si en la agenda sólo pudiéramos dar de alta registros y consultarlos, nuestro sistema sería muy rígido, ya que si una persona incluida en la agenda cambiara alguno de sus datos, por ejemplo, su dirección, tendremos que darle de alta de nuevo introduciendo todos sus datos.

Evidentemente, en los sistemas de gestión de bases de datos existen métodos para evitar duplicar información. En el caso del dBASE III podríamos hacerlo de dos formas distintas.

La primera de ellas es con la instrucción DELETE, que nos permite borrar un registro o un conjunto de ellos. Este borrado no es un borrado físico del archivo, sino que simplemente el registro o registros quedan marcados en el archivo como borrados, siendo posible recuperarlos más tarde con la instrucción RECALL. De esta forma podríamos introducir el nuevo dato sin que apareciera el registro duplicado.

Si lo que queremos es cambiar uno o varios campos del registro, pero sin eliminarlo de la base de datos, disponemos para ello de la función EDIT que nos permite editar y modificar un registro ya existente de la base de datos. La forma en que aparece en la pantalla el registro para su edición es igual a la de la figura 3.4. Podemos movernos por los distintos campos con las teclas de cursor para modificarlos.

También la función BROWSE nos permite modificar registros a nuestra voluntad moviéndonos entre ellos de arriba a abajo en forma tabular.



PROGRAMA

Hasta ahora todas las operaciones con la agenda las hemos realizado de modo interactivo, esto es, escribiendo los comandos desde el teclado, y el dBASE III respondiendo de una forma u otra. Sin embargo, si debemos hacer una operación con la agenda que conste de varias operaciones elementales, y tenemos que repetir ésta muchas veces, entonces es conveniente «programar» esta operación de tal forma que el dBASE III repita, cada vez que lo necesitemos, todo el conjunto de operaciones elementales necesarias para realizarla.

Para realizar esta «programación» el dBASE III dispone de un lenguaje que nos permite hacer todo esto. En los programas que hagamos podemos

incluir todos los comandos para la gestión de la base de datos, más una serie de instrucciones de control del programa.

En el dBASE III un programa es un fichero en el disco que cuando es invocado con el comando DO (Haz) se carga en memoria y empieza a ejecutarse. Cuando el programa termina su ejecución desaparece de la memoria.

Por ejemplo, si necesitáramos averiguar cuándo es el día del cumpleaños de alguien que está en la agenda para felicitarle, podríamos hacerlo de una forma interactiva, pero si tuviéramos que hacerlo muchas veces sería un proceso aburrido.

Por ello vamos a mostrar cómo realizar esta operación mediante un programa, que es como si añadiéramos un nuevo comando que fuera «buscar cumpleaños».

El programa es el siguiente:

```
* PROGRAMA PARA FECHAS DE CUMPLEAÑOS *
SET TALK OFF
CLEAR
CLOSE DATA
STORE CTOD("01-01-01") TO FECHA
USE AGENDA INDEX APELLIDO
GO TOP
@ 1,1 SAY "INTRODUCE LA FECHA :" GET FECHA PICTURE "D"
READ
CLEAR
LOCATE FOR FECHA=FECHA_NAC
IF .NOT. EOF()
   @ 1,1 SAY "EN ESTA FECHA CUMPLE AÑOS "+NOMBRE+" "+APELLIDOS
ELSE
   @ 1,1 SAY " NADIE CUMPLE AÑOS EN ESA FECHA "
RETURN
```

Fig. 3.18.

Para manejarlo, suponiendo que el fichero donde está almacenado se llamara «PROGRAMA», y en este momento empezara a ejecutarse, tendríamos:

Primero nos pediría la fecha de la cual queremos saber si coincide con el cumpleaños de alguien.

INTRODUCE LA FECHA : 01-01-01

Una vez introducida la fecha el programa se encarga de buscar si hay algún registro que tenga esa fecha en el campo fecha de nacimiento y en ese caso pone en la pantalla:

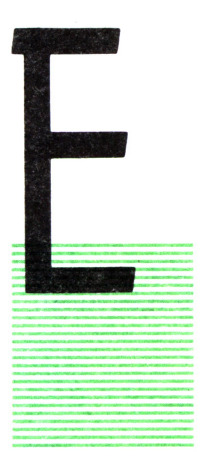
EN ESA FECHA CUMPLE AÑOS

seguido del nombre y los apellidos de la persona.

En caso contrario, si no existe nadie con esa fecha de nacimiento, aparecería en la pantalla la frase «NADIE CUMPLE AÑOS EN ESA FECHA».

Hasta ahora hemos visto las posibilidades de una base de datos concreta, que evidentemente tiene muchas más que se descubrirán al estudiarla con detenimiento.

ESTUDIO DE LAS BASES DE DATOS 4



FMS-80



EL FMS-80, creado por DJR Associates, de New York, es una base de difícil inclusión dentro de algún equipo.

No es relacional, ya que carece de las órdenes necesarias para manipular ficheros completos y combinaciones de éstos.

Veamos los requisitos hardware y algunas características:



Requisitos

Sistema operativo CP/M (MP/M, CDOS).
Sistema de microprocesador Z-80, 8080, 8085.
Un mínimo de memoria de 48 Kbytes.



Características

Número máximo de campos por registro: 255.
Número máximo de caracteres por campo: 254.
Número máximo de registros por fichero: 65.535

El FMS-80 es, en realidad, un paquete de dos niveles.

El primer nivel (el primero que se adquiere) es el FMS-81. Este es un sistema de fichero único, controlado por menú.

El segundo nivel (para cuando quiera ampliar la capacidad del FMS-81) es el FMS-82. Este permite el trabajo con ficheros múltiples, la ordenación

de los mismos e incorpora un lenguaje de programación (EFM). Este lenguaje no será difícil de aprender si conocemos el BASIC.

El EFM permite un trabajo mucho más potente con los datos almacenados así como hacer la búsqueda de algún elemento de modo más rápido.

El FMS-80 no posee campos de seguridad, pero mediante el uso del EFM se puede conseguir otro medio de salvaguarda.

La transferencia a/o desde ficheros que no sean del FMS-80 no se puede hacer normalmente; pero con pequeñas reducciones no dará problemas a los programas escritos en COBOL y PL/1; con el BASIC sí los dará.

No posee capacidad multiusuario.

En cuanto a las velocidades, el tiempo de acceso es de 1 a 20 seg. (dependiendo del fichero); y la velocidad de procesamiento es de 2500 registros por minuto.

El acceso a los datos puede hacerse de forma secuencial e indexada.



DBASE II

El dBase (tanto II como III) es una base de datos de tipo relacional.

Se compone de dos grupos de ficheros:

— los que contienen las órdenes ejecutables.
— los que contienen la información.

A continuación repasaremos los requisitos y características de esta BD:



Requisitos

Microprocesador 8080, 8085, Z-80
Memoria mínima 48K
Sistema operativo CP/M 80 y 86; MS/DOS
Ordenador IBM PC y compatibles



Características

Número de campos por registro: 32
Número de caracteres por registro: 1000
Número de registros por fichero: 65535
Número de ficheros abiertos simultáneamente: 2

El dBase II puede usarse con órdenes de acción directa o con series de instrucciones, que reúnen todas las características de un auténtico lenguaje.

Veamos algunos de los ejemplos de cómo trabajan los comandos del dBase II.

Una ventaja que tiene esta base de datos es que posee una instrucción HELP que nos ayudará a comprender el significado y formato de otras órdenes.

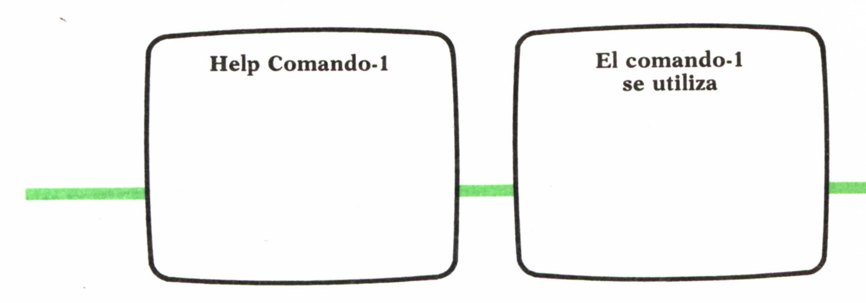


Fig. 4.1. Si no sabemos para qué sirve una orden tecleamos HELP seguido de la orden y nos aparecen sus usos.



OPERACIONES ELEMENTALES

Creación de ficheros

Una vez introducido el disco del dBase II, tecleamos DBASE y damos a la tecla RETURN. Cuando aparezca un punto en la pantalla podemos empezar a hacer lo que queramos. La orden para crear un fichero es CREATE. Si queremos crear un fichero con los datos de nuestra agenda:

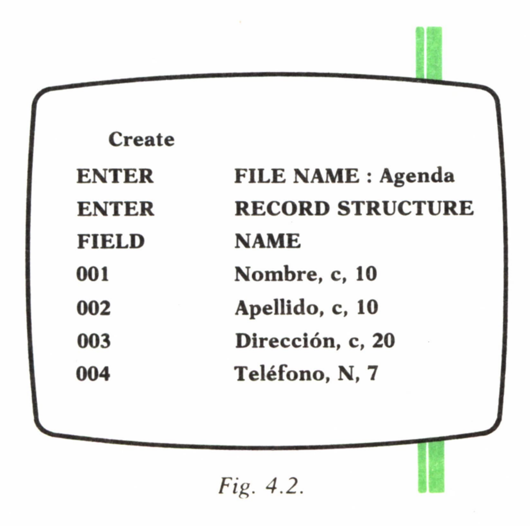


Fig. 4.2.



Introducción de información

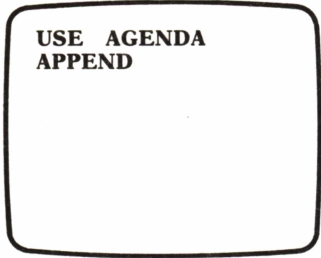




Fig. 4.3.

Para empezar a introducir datos tecleo USE y doy a la tecla RETURN; luego tecleo APPEND.

La orden USE sirve para llamar a un archivo con el que se va a operar. APEND nos permite introducir datos.



Listado de registros

Puedo hacerlo mediante el comando DISPLAY. Su formato es:

	ALL		
DISPLAY	RECORD N	OFF	FOR <expressión>
	NEXT M		

Mediante ALL: visualizamos todos los registros; con RECORD N visualizo el N-ésimo registro; NEXT M visualiza los M siguientes al que está activo. El OFF sirve para no visualizar el número del registro/s.

FOR <expresión> lista los registros que cumplan la condición expresada a continuación del FOR. Siguiendo con el ejemplo de la agenda, veamos los registros que hemos introducido.

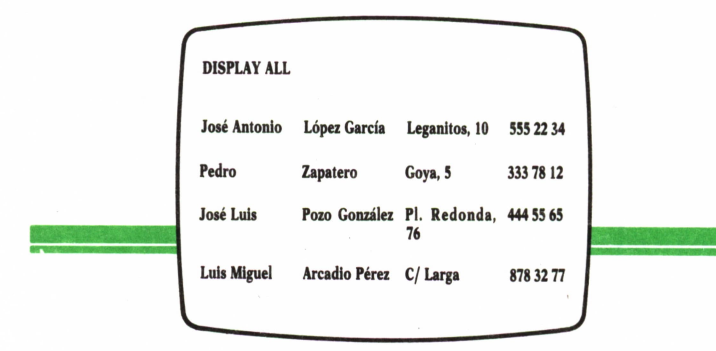
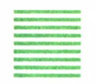


Fig. 4.4.



Actualización de información

Con **APPEND** incluimos otro registro. Con **DELETE** los borro. Su formato es parecido al de DISPLAY.

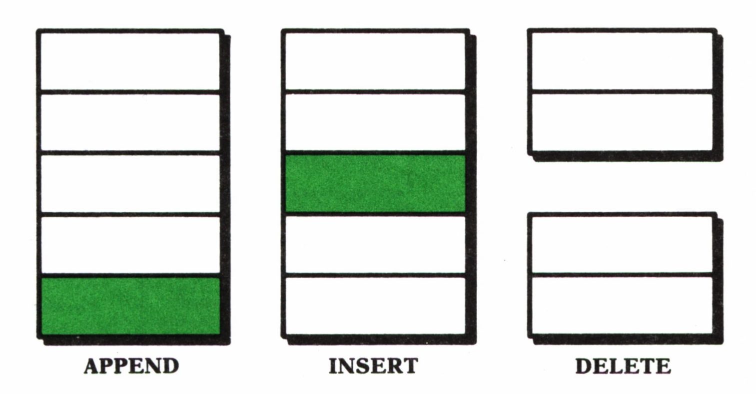


Fig. 4.5. El append incluye registros después del último, mientras que insert lo hace en cualquier parte.



Otras órdenes

Una bastante importante es JOIN. Necesita de la existencia de dos ficheros. Su formato es:

JOIN TO <NUEVO FICHERO> FOR <EXPRESSION> FIELDS

El dBase II permite la ordenación de más registros. Para ello tenemos. la instrucción **SORT** que crea un nuevo fichero a partir de uno viejo, en el que los registros del fichero original son ordenados físicamente según el valor de un campo elegido como clave de ordenación.

Lo que hace es unir dos ficheros creando uno nuevo cuyos campos son una combinación de cualquiera de los campos originales.

Aunque existen otros comandos no es misión de este libro profundizar en el estudio concreto de una base de datos; con lo que hemos dicho basta para hacerse una idea de cómo funciona el dBASE III (en modo comandos).

Numero	Nombre	Número de departamento
1	Juan	20
2	José	30
3	Pedro	20

Número de departamento	Función
20	Ventas
30	Dirección

JOIN por número de departamento

Número	Nombre	Número de departamento	Función
1	Juan	20	Ventas
2	José	30	Dirección
3	Pedro	20	Ventas

Fig. 4.6. Los ficheros que van a ser unidos necesitan tener algún campo en común (en este caso el número de departamento).

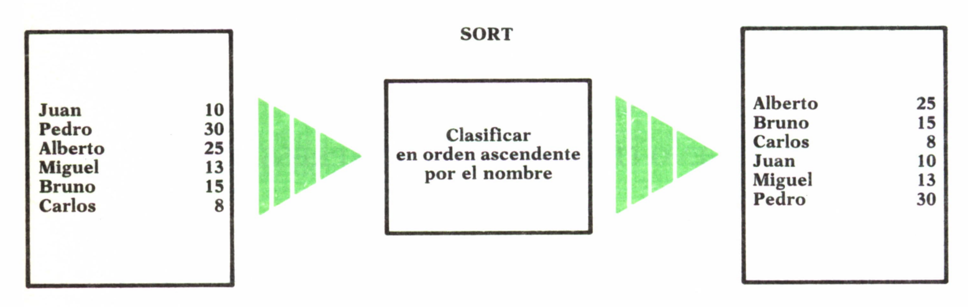


Fig. 4.7.

Si quiere seguir profundizando en esta base de datos le remitimos al Manual o a alguno de los libros que sobre esta base de datos hay.

Las velocidades de trabajo son:

— Tiempo de acceso: el acceso indexado (mediante campo clave) es de 0,5 seg. en disco rígido y 2 seg. con disquette.

— Velocidad de procesamiento: 10580 registros por minuto en disco rígido.

A continuación veamos algunas características del dBase II:

— Puede usar ficheros múltiples.
— No posee menús, lo cual puede complicar su uso por parte de personas no iniciadas.
— Controlado por órdenes y programas.
— No posee capacidad de ser utilizado por más de un usuario a la vez.
— No dispone de campos de seguridad.
— El acceso a los datos puede realizarse de modo secuencial, directo o indexado.
— Posee una buena documentación. Aparte del Manual que facilita el fabricante, existen bastantes estudios sobre esta base de datos en el mercado.

Conclusión: El dBase II es una de las BD más usadas y vendidas. Seguramente le será recomendada cuando vaya a comprarse una. Su uso está muy extendido y eso dice bastante a su favor.



R: BASE 5000

EL R:base 5000 es una base de datos de tipo relacional que almacena los datos en columnas y filas estructuradas en forma de tabla.

Veamos los requisitos hardware que necesita:

— Ordenador IBM PC y compatibles 100%.
— Sistema operativo MS/DOS.
— Memoria mínima 237 Kbytes.

Sus características físicas principales:

— Número máximo de campos por registro: 400.
— Número máximo de caracteres por registro: 1350.
— Número máximo de registros por fichero: No tiene límite hasta 19 Mbytes (que es el tamaño máximo del fichero permitido por el MS/DOS.)
— Número máximo de ficheros que maneja: 40.

El R:base 5000 es una de las bases de datos más completa y potente que puede adquirirse hoy en el mercado.

La firma creadora lanzó el R:Base 4000 con el propósito de competir con la popular dBASE II. Pero la R:base carecía de un lenguaje de programación para crear aplicaciones más potentes, que impidió que tuviera más éxito.

El R:base 5000, por el contrario, ha corregido la mayoría de los defectos de su predecesor.

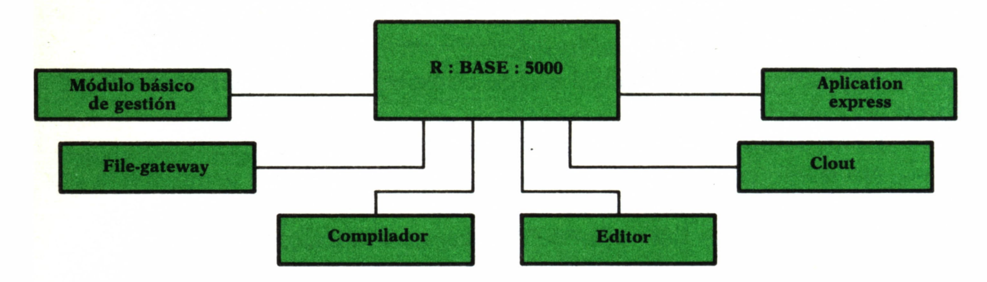


Fig. 4.8. Partes del R. base 5000.

El programa se suministró en 6 disquettes que contienen:

— Módulo básico del sistema de gestión.
— Un programa llamado FILEGATEWAY, que permite la transferencia de información desde otras bases de datos, creando automáticamente un fichero con la información transvasada.
— Un compilador.
— Un editor.
— Un generador de aplicaciones, llamado «Aplication Express».

Este sistema dirigido por menús automatiza y dirige al usuario a través del proceso de diseño y modificación de los ficheros.

Este programa es una de las principales ventajas que ofrece esta base de datos sobre el dBASE III y el KNOWLEDGEMAN, ya que además de reducir sustancialmente el tiempo requerido para desarrollar una aplicación hace posible que la misma sea realizada por usuarios sin conocimientos ni experiencia en programación.

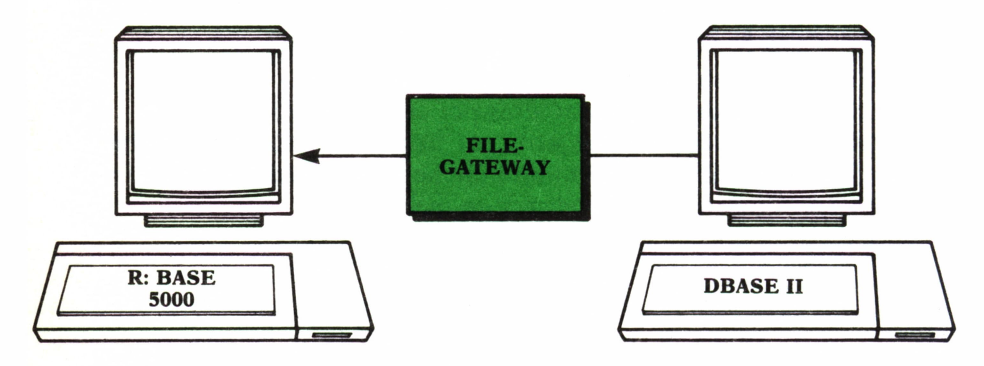


Fig. 4.9. Mediante el Filegateway puede transferir información de otras bases de datos.

El R:Base 5000 también incorpora un lenguaje de programación. Este lenguaje no posee tantas instrucciones como el del dBASE III, pero es suficiente para la mayoría de las aplicaciones que se nos puedan ocurrir.

Dispone de 80 instrucciones para aceptar información, imprimir, manipular ficheros y controlar el entorno del sistema.

Sólo dispone de las cuatro operaciones aritméticas básicas, suma, resta, multiplicación y división.

Para escribir los programas se utiliza el editor de texto RBEDIT.

El R:base 5000 es también una buena herramienta para sacar informes.

Estos pueden incluir 10 niveles de subtotales, contener información de 40 ficheros distintos y calcular hasta 40 variables a partir de los datos contenidos en el informe.

El proceso de creación de informes está controlado por menús.

En primer lugar, se crean los encabezamientos del informe, los piés de página, los encabezamientos de cada columna y cualquier otra información referente a la composición del texto.

Una vez diseñada la estructura básica se señalan las posiciones en que serán impresas las variables (hasta un máximo de 40).

Las principales ventajas del generador de informes son:

— Facilidad de uso.
— Buen manejo de subtotales.
— Acceso a la información de bastantes ficheros.

Aunque el R:base 5000 no lo incluye, es posible adquirir el módulo CLOUT (Conversational Language Option) concebido para interrogar a la base de datos en lenguaje natural, de forma que la persona que realiza la consulta no tiene por qué saber la sintaxis del lenguaje con que trabaja el R:base 5000, ni la estructura de la base de datos.

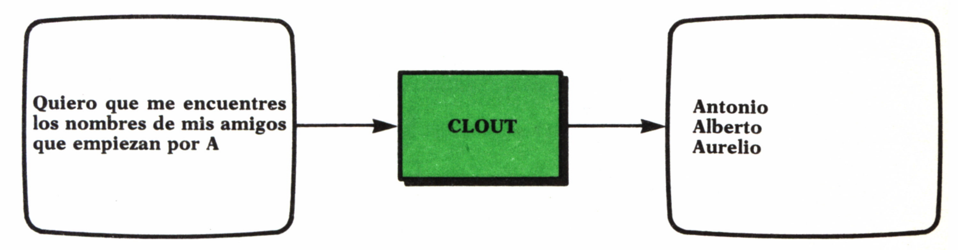


Fig. 4.10. El Clout traduce las órdenes del lenguaje normal a lenguaje entendido por la máquina.

Así, por ejemplo, es posible hacer el siguiente requerimiento: «Show me all books, whose price is greater than 200 pesetas» («Enséñame todos los libros cuyo precio es superior a 200 pesetas»).

Hay que anotar que el lenguaje natural para el CLOUT es el inglés y no el español.

Otras ventajas que encontramos en el R:base 5000 son:

— Permite el acceso de varios usuarios.

— Pueden definir «claves» (claves de entrada) para ficheros o usuarios, asegurando así la integridad de la información contenida en los ficheros.



Corolario

El R:base 5000 es una base de datos destinada a aquellos usuarios que desean desarrollar por sí mismos sus aplicaciones y no disponen de conocimientos de programación.

Aquellos que hayan trabajado con el dBASE III encontrarán muchas similitudes entre el lenguaje de esta base de datos y la del R:base 5000.



PFS FILE-REPORT

Como revelan sus «apellidos» (File-report), el PFS consta de aspectos: gestión de ficheros (lo que hemos venido llamando base de datos) y la producción de informes.

Estos dos aspectos son independientes entre sí, pudiéndose adquirir por separado.

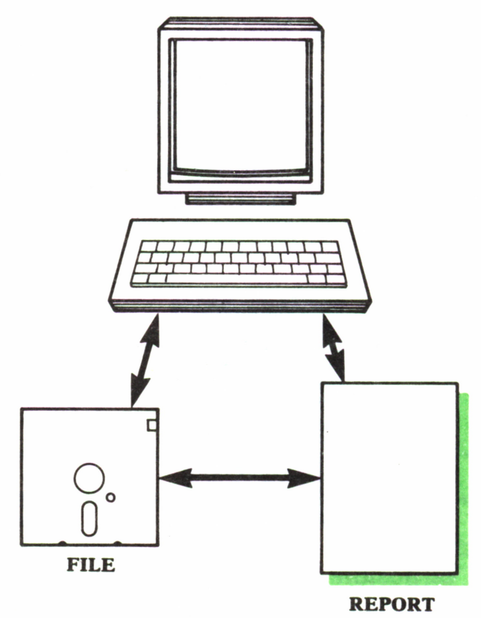


Fig. 4.11. Dentro del PFS hay dos partes, el PFS-Tele y el PFS-Report, que pueden trabajar por separado, pero que serán más eficientes si lo hacen juntos.

PFS-FILE

Se trata de un programa destinado a almacenar datos en ficheros y recuperarlos cuando se requiera.

En general el PFS-File puede realizar tres procesos:

— Diseño de la estructura con la que se guardarán los datos.
— Introducción de los datos.
— Recuperación de los datos en cualquier orden (que no tiene que ser en el que se introdujeron).

Veamos algo de su funcionamiento:

Después de introducir el disco con el programa PFS-File en la unidad correspondiente aparece un menú que proporciona 3 opciones:

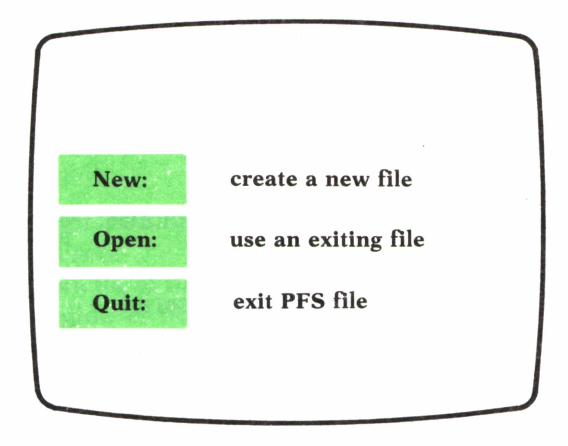


Fig. 4.12.

1. La opción NEW sirve para abrir nuevos ficheros. Después de seleccionarla, elegimos entre las opciones OK (confirma lo que hemos hecho antes) y CANCEL (lo anula).

Después aparecerá un mensaje de «FILE IN PAGE 1 OF DESIGN», que nos indica que debemos definir los campos de los que van a constar los registros.

Teniendo la estructura de los registros podemos dedicarnos a introducir la información seleccionando la opción adecuada.

2. La opción OPEN permite abrir un fichero ya creado para manipular sus datos. Permite la ordenación.

3. La opción QUIT provoca el abandono del programa retornando el control al sistema operativo.

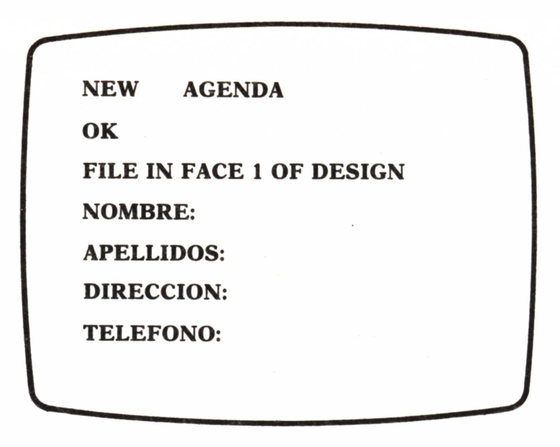


Fig. 4.13. Proceso de creación de un fichero y definición de sus campos.



PFS-REPORT

El PFS-Report tiene básicamente tres funciones:

— Escribir un informe.
— Crear el formato con el que se debe escribir.
— Cambiar las cabeceras de página.

Los ficheros a tratar por el PFS-Report tienen que haber sido previamente creados por el PFS-File.

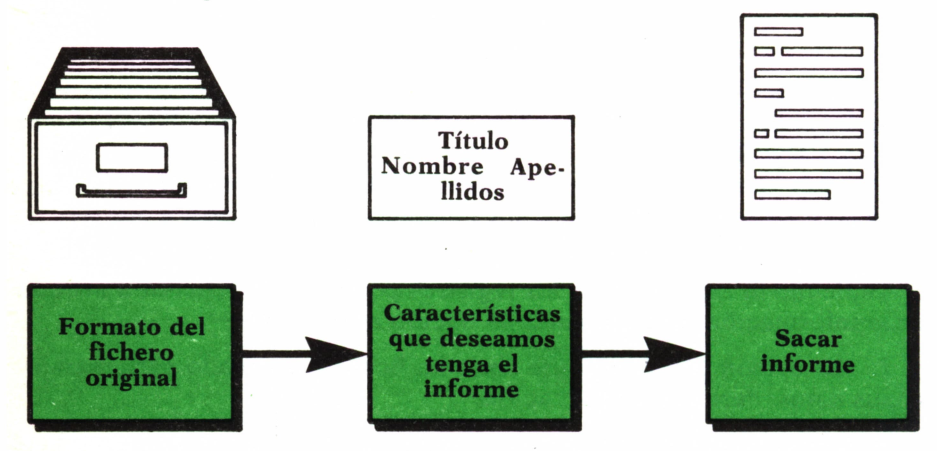


Fig. 4.14. Proceso para producir un informe.

Una vez invocado el programa PFS-Report, aparece un menú en la pantalla aportando dos alternativas.



Fig. 4.15.

1. OPEN

Después de elegir esta opción aparecen en la pantalla los ficheros almacenados en el disco.

El programa solicita el nombre del fichero que va a ser procesado.

Después de elegido el fichero aparece un menú con cuatro opciones (cuatro nuevos menús).

— FILE: Llama al fichero del que sacar un informe.
— EDIT: Facilita el trabajo de edición.
— FUNCTION: Selecciona algunas de las funciones que ofrece el PFS-Report para producir informes.
— FORMATS: Establece los formatos de escritura.

2. QUIT

Finaliza la sesión de trabajo con el PFS-Report.

Veamos a continuación los pasos que seguiremos para producir un informe.

Es necesario usar 3 menús:

— Indicar el formato en que están almacenados los datos en el fichero del que vamos a sacar el informe.
— Definir las características generales del informe (título, tamaño de página).

— Indicar los datos que deben aparecer en el informe. Puede que nos interese que no todos los registros aparezcan en el informe, sino sólo los que cumplen ciertas condiciones.

Supongamos que tenemos un fichero llamado agenda donde guardamos el nombre, apellidos, dirección y teléfono de nuestros conocidos y queremos hacer un informe o simple listado de todos ellos y sus datos.

Después de arrancar el PFS-Report y teclear la opción OPEN AGENDA, solicitamos el menú de funciones y dentro de él seleccionamos la función «PRINT A REPORT» (escribir un informe).

Como vamos a visualizar en el informe a todos nuestros conocidos, no hay que especificar ninguna condición.

A continuación aparece el menú que nos permite definir las características del informe (líneas por página, título del informe).

Después especifica los formatos de salida (orden de salida, columnas para cada campo, etc.).

La clasificación se produce sólo si en el disco hay espacio suficiente para crear un nuevo fichero donde ordenar los datos según el valor de algún campo. Puede que especifiquemos al PFS-Report la opción de ordenación y nos aparezca el mensaje «The disk is full», en cuyo caso nuestra orden queda sin sentido.

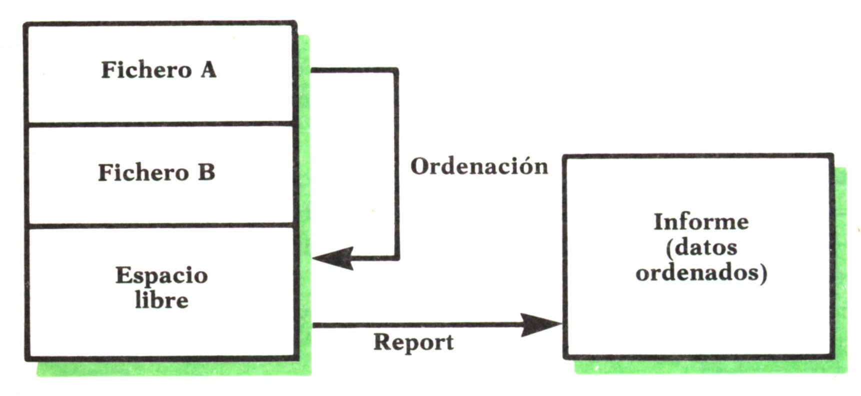


Fig. 4.16. La ordenación requiere un espacio libre en el disco para poder ejecutarla.

Después de esto ya podemos mandar la orden a la impresora para obtener nuestro informe.



CONDOR

Se trata de una base de datos de tipo relacional controlada por órdenes y menús.

Sus órdenes operan considerando los campos como un todo, esto es, no hay posibilidad de acceder a campos de registros individuales.

Veamos un ejemplo de esto.

Si tuviésemos un campo de nuestro fichero de productos, que se llamara PRECIO, no podría modificar el precio de un registro particular. Al acceder al campo «PRECIO» accedo genéricamente al PRECIO de todos los productos y al modificar uno los modifico todos.

A continuación veamos los requisitos hardware y algunas características.

Los requisitos son:

— Sistema microprocesador Z-80
— El sistema operativo CP/M
— Memoria mínima 48 K

Las características:

— Número máximo de campos por registro: 127.
— Número máximo de caracteres por campo: 127.
— Número máximo de registros por fichero: 32767.

Una de las ventajas del CONDOR es que permite el trasvase de información tanto desde programas como de otras bases de datos. No tiene problemas en aceptar programas escritos en BASIC y COBOL, ya sea para leer o escribir.

El Cóndor se vende en tres niveles. El nivel 1 es un sistema de gestión de ficheros con capacidad multifichero.

El nivel 2 incorpora los verdaderos operadores relacionales que la convierten en una auténtica base de datos relacional.

El nivel 3 añade un generador de informes y un esquema de indexación.

El generador de informes está controlado por menús (la única parte).

Otros aspectos que cabe reseñar del Cóndor son:

— No posee capacidad para crear palabras o códigos de seguridad.
— Utiliza ficheros múltiples.
— Sólo permite clasificar fichero de menos de 128 K.
— No puede ser usado por más de un usuario simultáneamente.
— Posee buena documentación.

Hay bastantes estudios hechos sobre esta base de datos.

En cuanto a la velocidad máxima de procesamiento es de 23.000 registros por minuto en disco rígido.

dBASE III

Es una base de datos de tipo relacional.

Veamos los requisitos hardware y algunas características:

- Microprocesador 8088, 8086.
- Memoria mínima: 256 Kbytes.
- Sistema operativo: MS-DOS, PC-DOS 2.0 (y superiores), UNIX.

Las características:

- Número máximo de campos por registro: 128.
- Número máximo de caracteres por registro: 4000.
- Número máximo de registros por fichero: 1.000.000
- Número máximo de ficheros abiertos simultáneamente: 10.

La introducción del dBASE II en el mercado supuso un gran adelanto en el acercamiento de las bases de datos a los ordenadores personales.

Aunque fue bastante bien admitida en medios informáticos también recibió críticas. Estas se debieron a su lenta velocidad de proceso y reducido tamaño de bases de datos que admitía. Los no programadores se quejaban del difícil lenguaje de programación (y consiguientemente de su uso) que soportaba.

El dBASE II trabajaba con microprocesadores de 8 bits; el dBASE III lo hace con uno de 16.

El microprocesador de 16 bits tiene la ventaja sobre el de 8 de que procesa los datos más de prisa; además, su uso es el que se está extendiendo por todas partes.

Aunque desarrollado a partir del dBASE II, el dBASE III es totalmente distinto de su predecesor.

Las principales diferencias son :

- Utiliza microprocesador de 16 bits.
- La velocidad de proceso es mayor.
- Tiene mayor capacidad de memoria.
- Es más fácil familiarizarse con él.

Una de las deficiencias del dBASE II era su difícil comprensión, ya que no trabajaba con menús.

El dBASE III incorpora menús, con lo que es relativamente fácil familiarizarse con él en poco tiempo.

Sin embargo, dada la potencia del lenguaje que incorpora, es difícil llegar a manejarlo en profundidad sacándole todo el partido posible.

La información en esta base de datos puede ser procesada de tres formas:

La ya mencionada de menús.

Una forma de manipular la información en un fichero de datos es el método de proceso iterativo de instrucciones (órdenes). Consiste en introducir las órdenes específicas de dBASE III a través del teclado. Después de introducida la orden, los resultados se visualizan en la pantalla.

Otro método de procesar la información en dBASE III es el proceso de instrucciones por lotes (lo que antes hemos llamado programas). Las operaciones a efectuar se definen previamente como un conjunto de instrucciones que posteriormente se ejecutan todas seguidas. Estas instrucciones se guardan en un fichero de instrucciones que puede ser considerado un programa de ordenador.

Las operaciones que componen el programa pueden seleccionarse mediante menús.

Una vez visto cómo procesa el dBASE III las operaciones, vamos a estudiar una característica un poco especial de esta base de datos: posee nueve tipos distintos de ficheros según la naturaleza de la información que guarda cada uno.

Estos son:

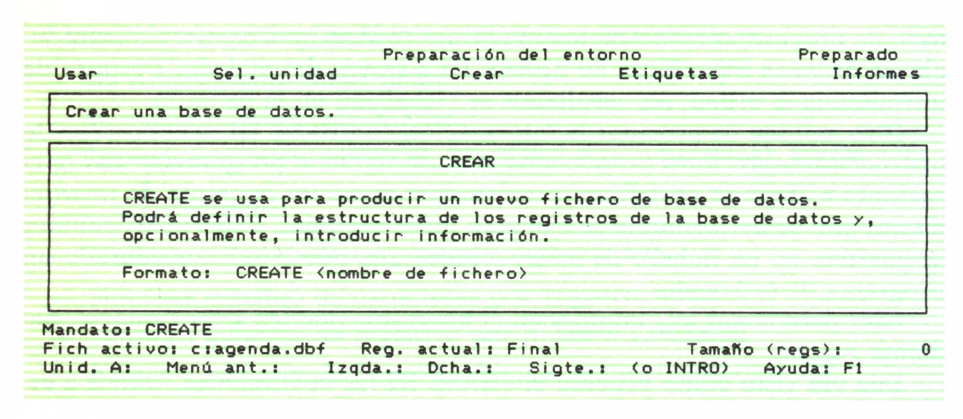
1. Fichero de base de datos. Contiene los registros y campos donde guardamos nuestros datos.
2. Ficheros de datos memo. Almacenan grandes bloques de datos.
3. Ficheros índice. Ofrecen el espacio necesario para la ordenación de algún fichero. Así puedo acceder a los datos, en un orden distinto al que originariamente poseían.
4. Fichero de instrucciones. Almacenan el conjunto de instrucciones que van a ser procesadas.
5. Fichero de formato. Almacena información que especifica el formato de salida de los registros.
6. Ficheros de etiquetas. Almacenan información sobre las etiquetas que se van a imprimir.
7. Ficheros de memoria. Almacenan el contenido de las variables de memoria. Estas representan las direcciones temporales de memoria que pueden guardar resultados de cálculos utilizables en procesos posteriores.
8. Ficheros de formato de informes. Contienen información para generar informes. Estos los obtendré mediante la instrucción REPORT.
9. Ficheros de texto de salida. Almacenan texto que puede ser compartido con otros programas de ordenador. Permiten las conexiones necesarias para el intercambio de información entre el dBASE III y el exterior.

Veamos a continuación un breve ejemplo de cómo trabaja el dBASE

III con menús, para una aplicación sencilla como puede ser almacenar los datos de nuestra agenda.

Lo primero es crear el fichero AGENDA y la estructura de los registros.

Seleccionamos el menú CREAR.



```
                         Preparación del entorno                Preparado
Usar         Sel. unidad        Crear          Etiquetas           Informes

 Crear una base de datos.

                                CREAR

      CREATE se usa para producir un nuevo fichero de base de datos.
      Podrá definir la estructura de los registros de la base de datos y,
      opcionalmente, introducir información.

      Formato:  CREATE <nombre de fichero>

Mandato: CREATE
Fich activo: c:agenda.dbf   Reg. actual: Final        Tamaño (regs):      0
Unid. A:   Menú ant.:   Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

Fig. 4.17.

y tecleamos

```
c:agenda.dbf                                        Bytes restantes:   3933
                                                    Campos definidos:     4
   Nomb campo   Tipo       Ancho  Dec          Nomb campo   Tipo      Ancho  Dec

 1 NOMBRE       Car/texto   20
 2 APELLIDOS    Car/texto   20
 3 DIRECCION    Car/texto   20
 4 TELEFONO     Numérico     7     0
```

Fig. 4.18.

Una vez definida la estructura introducimos los datos; para ello seleccionamos el menú AÑADIR.



```
                         Modificar base de datos                Preparado
Añadir     Hojear     Editar     Marcar     Desmarcar     Sustituir    Posición

 Añadir registros a la base de datos en uso.

                              AÑADIR

      Con APPEND podrá añadir información al fichero de base de datos usando
      la edición interactiva en la pantalla. La información se añade de
      registro en registro, y de campo en campo.

      Formato: APPEND [BLANK / FROM <nombre de fichero>
               [FOR / WHILE <condición>] [SDF / DELIMITED]]

Mandato: APPEND
Fich activo: c:agenda.dbf   Reg. actual: Final        Tamaño (regs):      0
Unid. A:   Menú ant.:   Izqda.:  Dcha.:   Sigte.:  (o INTRO)   Ayuda: F1
```

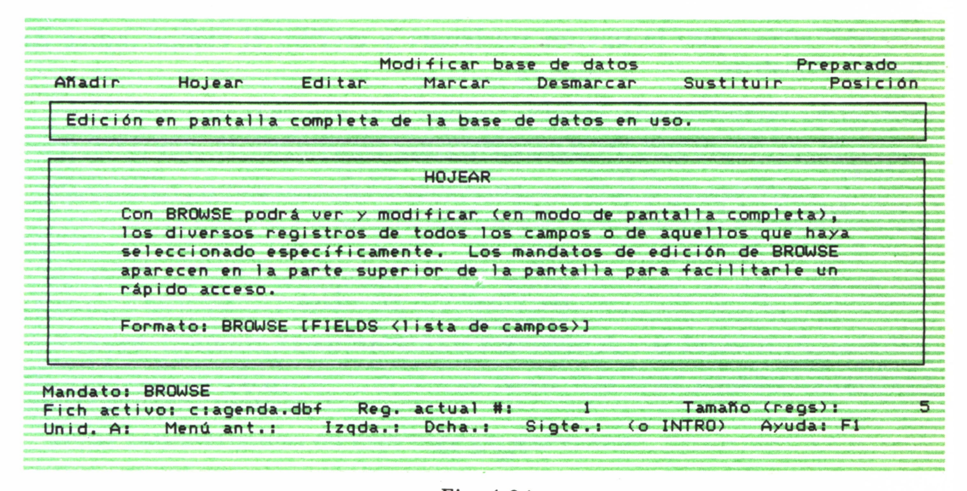
Fig. 4.19.

Y empezamos a teclear los datos de nuestros amigos.

```
Reg. Nº         1
NOMBRE          Jose Antonio
APELLIDOS       Lopez Garcia
DIRECCION       C/ leganitos  10
TELEFONO        5552234
```

Fig. 4.20.

Si quisiéramos ver los registros que hemos introducido seleccionamos, dentro del menú **MODIFICAR**, la opción **HOJEAR**.



```
                              Modificar base de datos                 Preparado
Añadir     Hojear     Editar     Marcar     Desmarcar     Sustituir     Posición

 Edición en pantalla completa de la base de datos en uso.

                                     HOJEAR

      Con BROWSE podrá ver y modificar (en modo de pantalla completa),
      los diversos registros de todos los campos o de aquellos que haya
      seleccionado específicamente.  Los mandatos de edición de BROWSE
      aparecen en la parte superior de la pantalla para facilitarle un
      rápido acceso.

      Formato: BROWSE [FIELDS <lista de campos>]

Mandato: BROWSE
Fich activo: c:agenda.dbf   Reg. actual #:     1      Tamaño (regs):       5
Unid. A:   Menú ant.:    Izqda.:  Dcha.:    Sigte.:  (o INTRO)   Ayuda: F1
```

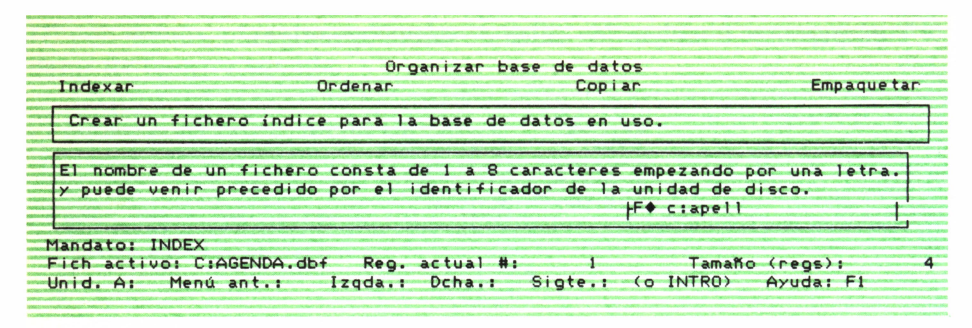
Fig. 4.21.

Y obtenemos

```
Reg. Nº              1     agenda
NOMBRE-------------- APELLIDOS----------- DIRECCION----------- TELEFONO
Jose Antonio         Lopez Garcia         C/ leganitos  10      5552234
Pedro                Zapatero             C/ GOYA         5     3337812
Jose  Luis           Pozo Gonzalez        Pl/ Redonda    76     4445565
Luis Miguel          Arcadio Perez        C/ Larga     255      8783277
```

Fig. 4.22.

Si en nuestros planes futuros está la ordenación por algún campo índice, tendremos que definirlo. Para ello seleccionamos el menú **ORGANIZAR** y la opción «indexar» indicando qué campo de los que constan nuestros registros va a ser el campo-índice (elegimos el campo apellido).



```
                         Organizar base de datos
Indexar                 Ordenar                 Copiar              Empaquetar

Crear un fichero índice para la base de datos en uso.

El nombre de un fichero consta de 1 a 8 caracteres empezando por una letra.
y puede venir precedido por el identificador de la unidad de disco.
                                               F♦ c:apell

Mandato: INDEX
Fich activo: C:AGENDA.dbf    Reg. actual #:       1        Tamaño (regs):        4
Unid. A:    Menú ant.:    Izqda.:   Dcha.:    Sigte.:   (o INTRO)    Ayuda: F1
```

Fig. 4.23.

Si lo que queremos es ordenarlos por apellido, dentro del menú **ORGANIZAR** selecciono la opción «ordenar» y escribo el campo-índice de nuestros registros.

```
                         Organizar base de datos
Indexar                 Ordenar                 Copiar              Empaquetar

Introduzca la expresión de la clave:  apellidos

                               Nomb. campo    Tipo campo      Ancho    Nº Dec
                               NOMBRE          Car/texto       20
                               APELLIDOS       Car/texto       20
                               DIRECCION       Car/texto       20
                               TELEFONO        Numérico         7       0
Mandato: INDEX
Fich activo: C:AGENDA.dbf    Reg. actual #:       1        Tamaño (regs):        4
```

Fig. 4.24.

Los resultados serán.

```
Reg. Nº             4     AGENDA
NOMBRE-------------- APELLIDOS----------- DIRECCION----------- TELEFONO
Luis Miguel          Arcadio Perez        C/ Larga      255     8783277
Jose Antonio         Lopez Garcia         C/ leganitos   10     5552234
Jose  Luis           Pozo Gonzalez        Pl/ Redonda    76     4445565
Pedro                Zapatero             C/ GOYA         5     3337812
```

Fig. 4.25.



REVELATION

Revelation es otra de las bases de datos, ha sido creada para micros, de tipo relacional.

Veamos los requisitos que necesita para funcionar, así como sus características:



Requisitos

— Consumo mínimo de memoria: 320 K. Es mejor que esté soportada por disco fijo que por disquette.
— Sistema operativo: MS-DOS o DOS (versión normal).
— Ordenadores: IBM PC y compatibles.



Características

— Número de campos por registro: 32.768.
— Número de registros por fichero: 30.000. Dependiendo de la capacidad del disco, el número de registros puede ser mayor.

Revelation puede trabajar con menús, órdenes y programas.

Los menús pueden ser suficientes para un usuario con necesidades medias.

Si se quiere profundizar en el manejo de esta base de datos habrá que saber usar el lenguaje de programación que incorpora el R/BASIC.

A mitad de camino entre el BASIC, el COBOL y el FORTRAN, el R/BASIC aumentó notablemente la potencia de esta base de datos, logrando unos resultados bastante buenos.

Otras características de esta base de datos son:

1. Su instalación no implica una renuncia a los «paquetes de software» más usados (Lotus 1-2-3, Symphony, Open Access) y permite el trasvase de información de ficheros de otra base de datos tipo dBASE II.
2. Tiene la posibilidad de asegurar campos (mediante códigos de seguridad) a intromisiones no deseables.
3. Permite el acceso multiusuario.
4. La velocidad de acceso y tratamiento es bastante buena.

Conclusión: Para sacarle el mayor partido posible a esta base de datos conviene estar un poco introducido en temas informáticos, aunque también ofrezca buenos resultados a los no iniciados.



SYMPHONY

Conviene aclarar que aunque estemos haciendo un estudio de las bases de datos, Symphony no es exactamente una de ellas.

Symphony es un paquete integrado.
¿Qué es un paquete integrado?
Someramente, podemos decir que es un único bloque que realiza funciones para las que normalmente se necesitan varios programas.
Así, el Symphony dispone de:

— Hoja de cálculo: permite realizar operaciones matemáticas, estadísticas y financieras, repetir automáticamente todos los cálculos si cambian los datos y hacer previsiones y proyecciones con mucha facilidad.
— Procesador de texto para escribir y modificar documentos.
— Gráficos: representa gráficamente datos numéricos.
— Base de datos: almacenar información, clasificarla y recuperarla según ciertos criterios.
— Comunicaciones para intercambiar información con otros ordenadores.

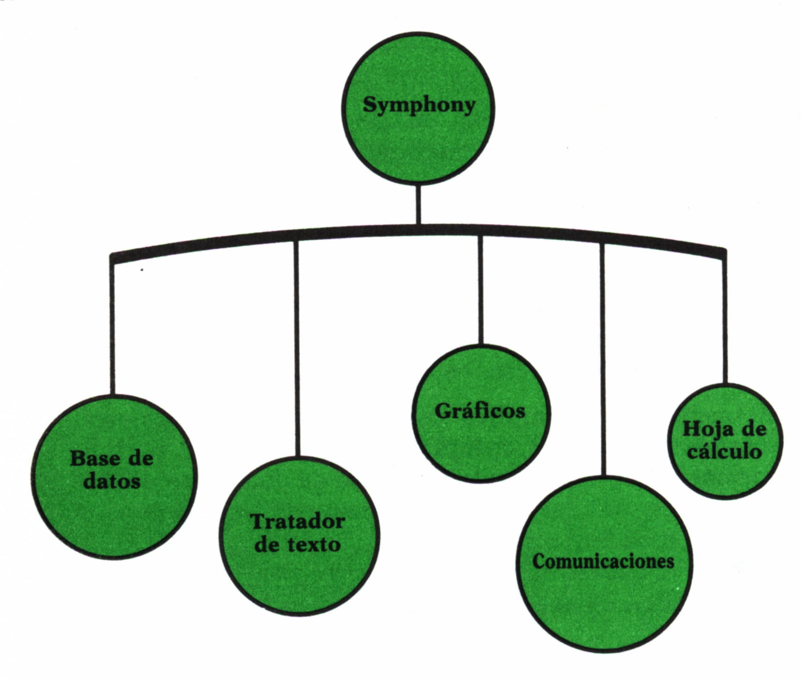


Fig. 4.26.

Existen muchos programas que realizan estas funciones por separado. La ventaja del Symphony es la integración de todas las funciones en un único paquete y la intercomunicación entre sus partes (por ejemplo: la hoja del cálculo permite usar información de la base de datos).
La integración tiene otra ventaja: sólo hace falta aprender a manejar un programa.

El Symphony es una herramienta de trabajo bastante potente; el llegar a conocerlo en profundidad es difícil, pero aprender lo imprescindible no es ningún problema, ya que incorpora un sistema de menús muy asequible.

Veamos algo de su modo de trabajo.

El Symphony está organizado en torno a la hoja de trabajo, que es una gran tabla, compuesta por celdas (8192 filas por 256 columnas).

Todas las operaciones se efectúan sobre ella.

Una hoja de trabajo tan grande no puede ser contemplada enteramente por el usuario.

Para subsanar esta dificultad usamos la «ventana».

Una ventana es una porción rectangular de nuestra pantalla a través de la cual vemos una parte de la hoja de trabajo.

Para definir la ventana seleccionamos la opción WINDOW (ventana) en el menú principal.

```
Modify current window or use another window                                    MENU
Window  File  Print  Configuration  Application  Settings  New  Exit
    A        B        C        D        E        F        G        H
1
2
```

Fig. 4.27.

Symphony dispone de cinco tipos distintos de ventanas a través de las cuales podemos ver los mismos datos de formas distintas.

Según el tipo de ventana que estemos usando, nos hallaremos en un determinado «entorno de trabajo», que determine las tareas que vamos a poder realizar.

Los entornos de trabajo son:

SHEET (Hoja): permite realizar operaciones de hoja de cálculo.
DOC (Texto): para trabajar con el procesador de texto.
GRAPH (Gráfico): para cuando deseemos obtener representaciones gráficas de los datos.
FORM (Base de datos): para realizar tareas en la base de datos.
COMM (Comunicación): para gestionar las comunicaciones con otros ordenadores.

Podemos seleccionar cualquiera de estos entornos con el menú que aparece al seleccionar la opción WINDOW, o sea, después del WINDOW, definimos el tipo de ventana (entorno de trabajo) con la que queremos trabajar.

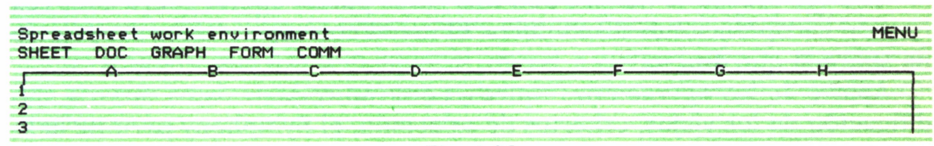


Fig. 4.28.

Ya que estamos en un libro de base de datos, elijamos la opción FORM que nos permite trabajar sobre la base de datos.

Una vez situados en esta opción delimitamos el tamaño y la situamos en la parte de la pantalla que queramos.

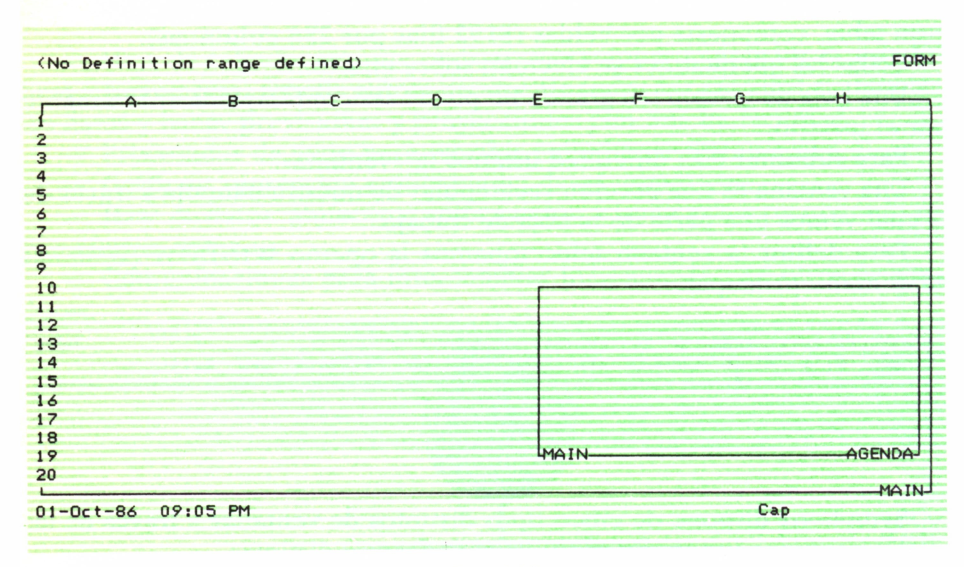


Fig. 4.29.

Una vez que estamos en el entorno de trabajo de la base de datos podemos realizar las operaciones típicas de cualquier base de datos nada más que eligiendo las opciones que nos ofrecen los menús.

Los menús de Symphony son un poco especiales. No ocupan toda la pantalla, sino sólo las dos o tres primeras líneas de la parte superior de la pantalla.

Siguiendo con las operaciones posibles de la base de datos, lo primero que debemos hacer es crear la estructura de los registros.

Para ello definimos en la primera línea los campos, con el tipo de carácter, y la longitud.

Luego, al introducir los datos en la base, aparecen los campos dentro de ventana requiriéndonos los datos.

Voy introduciendo en el fichero (ventana características) todos los datos que quiera.

Una vez creado el fichero con mis datos puedo operar sobre ellos realizando las operaciones disponibles. El menú correspondiente me posiciona sobre la que quiero.

Aparte de las operaciones típicas, como introducir datos nuevos, modificación de los viejos, criterios de selección, el Symphony permite la indexación y la ordenación de los datos.

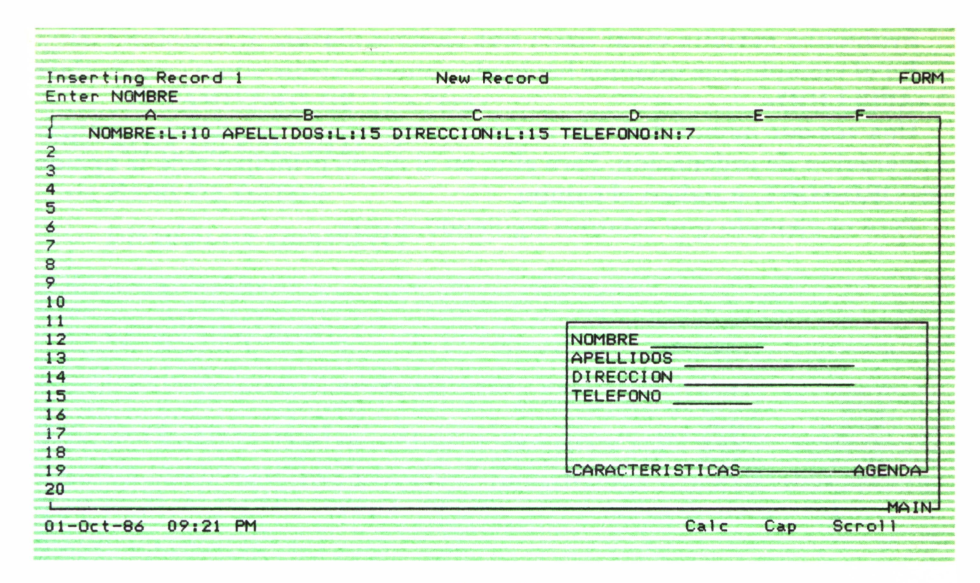


Fig. 4.30.

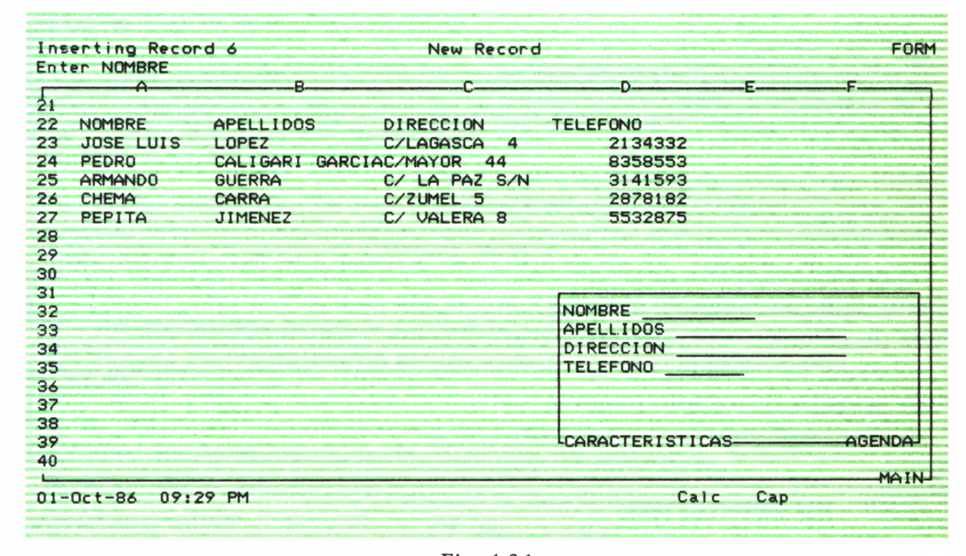


Fig. 4.31.

Veamos estas últimas operaciones con un poco más de detalle.

Lo primero que hay que hacer si vamos a querer ordenar nuestros datos es crear un campo índice que se tomará como referencia para la ordenación.

Volviendo al menú principal, elegimos la opción SETTINGS.

Symphony permite crear hasta tres campos claves.

En el ejemplo que estamos haciendo vamos a elegir como campo clave el «APELLIDO».

Una vez que hemos definido nuestro campo-clave, ya podemos realizar la operación de ordenación. Para ello, en el menú principal elijo la opción RECORD-SORT, e indico el campo.

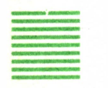
Para el ejemplo que estamos siguiendo queda:

```
A33:                                                                    SHEET
┌─────────A──────────────B──────────────C──────────────D──────────E──────────F─────┐
21
22  NOMBRE       APELLIDOS       DIRECCION       TELEFONO
23  PEDRO        CALIGARI GARCIAC/MAYOR  44               8358553
24  CHEMA        CARRA           C/ZUMEL 5                2878182
25  ARMANDO      GUERRA          C/ LA PAZ S/N            3141593
26  PEPITA       JIMENEZ         C/ VALERA 8              5532875
27  JOSE LUIS    LOPEZ           C/LAGASCA  4             2134332
28
29
30
31
32
33
34
35
36
37
38
39
40
└──────────────────────────────────────────────────────────────────────────MAIN┘
01-Oct-86  09:32 PM                                               Cap
```

Fig. 4.32.

Conviene volver a decir que además de las típicas operaciones que realiza toda base de datos, el Symphony puede operar con ellos mediante la hoja de cálculo, representarlos gráficamente, sacar informes, etc.

Esta es la verdadera potencia del Symphony, la integración de varias aplicaciones en un solo paquete.



FRAMEWORK

El FRAMEWORK es un paquete integrado que permite al usuario disponer de una forma rápida y eficaz de una serie de posibilidades para el desarrollo y gestión de ideas y aplicaciones.

A diferencia de otros paquetes integrados, aquí no se hace una distinción entre los distintos tipos de entornos, sino que todos los recursos y posibilidades del sistema son compartidas por todas las aplicaciones.

El FRAMEWORK dispone de:

— TRATAMIENTO DE TEXTOS
— HOJA DE CALCULO
— BASE DE DATOS
— DISEÑO DE GRAFICOS
— ESQUEMAS
— COMUNICACIONES
— LENGUAJE DE PROGRAMACION

En el FRAMEWORK es posible disponer de cada una de estas aplicaciones de forma múltiple; por ejemplo, se pueden tener simultáneamente varias hojas de cálculo independientes, varias bases de datos, etc.

Los requisitos del hardware son:

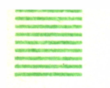
— ORDENADOR IBM PC CON MS-DOS
— MEMORIA RAM MINIMA DE 384 Kb
— DOS UNIDADES DE DISCO O UN DISCO Y UN DISCO DURO
— UN MONITOR

La idea fundamental de este paquete integrado es el concepto de «FRAME»; que se puede traducir por ventana. A diferencia de otros paquetes en los que las ventanas son simples zonas de visualización de la hoja de cálculo, aquí las ventanas son verdaderas estructuras de datos que permanecen aunque las borremos de la pantalla, y es dentro de estas ventanas donde se realizan las operaciones con los datos.

En el caso del FRAMEWORK, así como en la mayoría de los paquetes integrados, la base de datos es simplemente una tabla bidimensional de campos y registros que permanece en memoria, por lo que está limitada por el máximo de memoria RAM de la que dispongamos.

Su funcionamiento es muy sencillo, simplemente al crearla decimos el número de campos y registros de los que va a disponer y en ese momento se crea la base de datos, en la cual podemos introducir datos, realizar búsquedas, modificar, ordenar, etc., utilizando todos los mandatos del menú general.

Este paquete integrado tiene una ventaja adicional en el manejo de las bases de datos, y es que permite leer directamente ficheros producidos con el dBASE III sin necesidad de un programa traductor que cambie el formato de los datos.



EQUIPOS DOMESTICOS

NOMBRE : ARCHIVE

DESCRIPCION FUNCIONAL :

- Diseño de pantallas
- Búsquedas
- Ordenaciones
- Importación-exportación de datos

MICROORDENADOR : SINCLAIR-QL

NOMBRE : BASE DE DATOS

FUNCION : Base de Datos

DESCRIPCION FUNCIONAL :

- Búsquedas
- Ordenaciones
- Campos Operacionales
- Impresión de datos

MICROORDENADOR : SINCLAIR 128 K

NOMBRE : BASE DE DATOS

DESCRIPCION FUNCIONAL :

Incluye tres programas: micropen (para crear y mantener las bases de datos), repair (para actuar sobre el sistema) y perform (para diseñar formatos).

MICROORDENADOR : AMSTRAD CPC-128

NOMBRE : DATABOSC

DESCRIPCION FUNCIONAL :

Fichero o cuaderno electrónico en el cual, cada página tiene hasta 10 campos. Cualquier campo puede ser una combinación de caracteres por lo cual este programa puede guardar nombres, direcciones....

MICROORDENADOR : AMSTRAD CPC-464

NOMBRE : AMS BASE

DESCRIPCION FUNCIONAL :

- Se pueden crear archivos de datos de tal forma que el formato, temario y características las define el propio usuario. Se puede actualizar, consultar, listar, ordenar, sacar resultados, etc...

MICROORDENADOR : AMSTRAD CPC-464

NOMBRE : AGENDA 64

DESCRIPCION FUNCIONAL :

- Diseñado para poder servir de agenda personal.
- Llevará los nombres, direcciones, teléfonos,...

MICROORDENADOR : COMMODORE 64

NOMBRE : BIBLOS 64

DESCRIPCION FUNCIONAL :

- Permite llevar a su usuario el control de su biblioteca particular. Como opciones: altas,a bajas, autores, editoriales, etc.

MICROORDENADOR : COMMODORE 64

NOMBRE : DATAMASTER 64

DESCRIPCION FUNCIONAL :

- Totalmente programable en cuanto a formato de los ficheros, lo cual permite hacerla servir para muy diferentes tareas (archivos pedido, facturas, etc.).

MICROORDENADOR : COMMODORE 64

NOMBRE : FICHERO AGENDA

DESCRIPCION FUNCIONAL :

- Posibilidad de tener grabado en un fichero en cinta todas las direcciones, nombres teléfonos y observaciones de las personas que se desee.

MICROORDENADOR : COMMODORE 64

NOMBRE : FICHERO PERSONAL

DESCRIPCION FUNCIONAL :

- Permite al usuario mantener un fichero definido por él mismo, nombre y longitud de los campos, número total de fichas, etc.

MICROORDENADOR : COMMODORE 64

NOMBRE : FICHERO SELLOS (200 fichas)

DESCRIPCION FUNCIONAL

- Este programa ofrece la posibilidad de tener grabados en un fichero en cinta las referencias y valores de la colección de sellos clasificados por años y países.

MICROORDENADOR : COMMODORE 64

NOMBRE : FILOS 64

DESCRIPCION FUNCIONAL :

- Programa diseñado por filatélicos y coleccionistas de sellos y permite almacenar las colecciones ordenándolas por años y países.

MICROORDENADOR : COMMODORE 64

NOMBRE : NUMISMAT 64

DESCRIPCION FUNCIONAL :

- Permite llevar el control de las colecciones de monedas. Se pueden ordenar por años, panîses, precios, etc.

MICROORDENADOR : COMMODORE 64

NOMBRE : REFERENCIAS BIBLIOGRAFICAS

DESCRIPCION FUNCIONAL :

- Permite tener un disco, una relaciôn o un fichero referencias de artîculos o textos de revistas o libros. Se pueden hacer consultas por varios conceptos simultâneamente.

MICROORDENADOR : COMMODORE 64



EQUIPOS IBM-PC COMPATIBLES

NOMBRE : DOCUTEX

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 2 Unidades de diskette. Unidad de disco recomendable.

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos documental

DESCRIPCION FUNCIONAL :

- Ver descripciôn detallada

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : MICRONET, S.A..

NOMBRE : TEXTO

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 512 KB

MEMORIA DE MASA : Unidad de disco OBLIGATORIO

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Sistema de gestiôn documental diseñado para el almacenamiento y recuperaciôn de grandes volúmenes de informaciôn.

DESCRIPCION FUNCIONAL :

- Ver descripciôn detallada.

AMBITO DE APLICACION:

- General

MICROORDENADOR : IBM PC-XT/AT y compatibles

DOCUMENTACION :

PROVEEDOR : I + D. Informâtica y documentaciôn.

NOMBRE : OPEN-ACCESS

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 192 K

MEMORIA DE MASA : 2 Unidades de diskette.

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Paquete integrado con: hoja de câlculo, procesador de textos, grâficos 3-D, Comunicaciones y agenda.

DESCRIPCION FUNCIONAL :

- Ver descripciôn

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano/inglés

PROVEEDOR : SOFTWARE AND PRODUCTS INTERNATIONAL

NOMBRE : SYMPHONY

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 320 K RAM

MEMORIA DE MASA : 2 unidades de diskette.

SIST. OPERATIVO : DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Paquete integrado con: procesador de textos, gráficos, base de datos, comunicaciones y hoja de cálculo.

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION : -General

MICROORDENADOR : IBM PC-XT/AT y compatibles

DOCUMENTACION : Español, inglés

PROVEEDOR : CHIP ELECTRONICA

NOMBRE : FRAMEWORK

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 384 K

Unidad de disco: Opcional

MEMORIA DE MASA :

SIST. OPERATIVO : MS/DOS 1.2 ó superior

FUNCION : Paquete integrado con: hoja de cálculo, procesador de textos, base de datos, gráficos, comunicaciones y lenguaje de programación (FRED).

DESCRIPCION FUNCIONAL:

- Ver descripción detallada

AMBITO DE APLICACION:
- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano e inglês

PROVEEDOR : ASHTON-TATE

NOMBRE : INTEGRATED-7

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 320 K

MEMORIA DE MASA : 1 Unidad de diskette

SIST. OPERATIVO : DOS 2.0

SOFTWARE COMPLEMENTARIO :

FUNCION : Programa que combina, gestiôn de bases de datos relacional, correo electrônico, hoja electrônica, grâficos, tratamiento de textos, comunicaciones y emulaciôn determinada.

DESCRIPCION FUNCIONAL:

- El sistema de gestiôn de base de datos admite: 100.000 registros por fichero, claves, mûltiples, intercambio de datos con la hoja electrônica, consultas y bûsquedas por mûltiples campos,...

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Inglês

PROVEEDOR : SOFTRONICS, S.A.

NOMBRE : SERIE IBM ASSISTANT

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 K

MEMORIA DE MASA : 1 Unidad de diskette

SIST. OPERATIVO : DOS 1.1

SOFTWARE COMPLEMENTARIO :

FUNCION :

- Juego de programas que contiene: Filing assistant, reporting assistant, writing assistant, graphing assistant y planning assistant.

DESCRIPCION FUNCIONAL :

- El filing assistant convierte al PC en un fichero mecanizado por acceso fácil y rápido a los registros. El reporting assistant ayuda a transformar la información del Filing assistant en informes en plan profesional.

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC-XT/AT y compatibles

DOCUMENTACION : Inglés

PROVEEDOR : IBM

NOMBRE : SERIE PERFECT SOFTWARE II

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 K

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : DOS 2.00

SOFTWARE COMPLEMENTARIO :

FUNCION : Serie que comprende: perfect writer II, porfect calc II, perfect filer II.

DESCRIPCION FUNCIONAL :

- El perfect filer II es la base de datos de la serie. Permite intercambiar datos con la hoja electrónica, mailing con perfect writer o con otros procesadores de texto. Búsqueda por múltiples condiciones. Facilidades para la entrada automática de datos que se repiten en todos los registros.

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC-XT y compatibles

DOCUMENTACION : Inglés

PROVEEDOR : COMPULOGICAL, S.A.

NOMBRE : dBASE III

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 1 ô 2 unidades de diskette

SIST. OPERATIVO : MS-DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Sistema de gestiôn de base de datos relacional programable.

DESCRIPCION FUNCIONAL :

- Ver descripciôn detallada

AMBITO DE APLICACION :

- Todo tipo de empresas

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano, inglês u otro.

PROVEEDOR : ASHTON-TATE

NOMBRE : MANT 4.0

REQUERIMIENTOS DEL SISTEMA :

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 2 unidades de diskette.

SIST. OPERATIVO : DOS 1.1 ô posterior

SOFTWARE COMPLEMENTARIO : Gecom, Mantcopy

FUNCION : Base de datos con la cual se puede almacenar, clasificar y analizar toda la informaciôn deseada de forma râpida y sencilla.

DESCRIPCION FUNCIONAL :

- Control de inventarios
- Mantener y actualizar los clientes, productos, productores, representantes, etc.
- Conexiôn con los programas Mantlist, Mantmail, Mantcopy, Mantetiq y Gecom

AMBITO DE APLICACION :

- Cualquier empresario, gestor, administrador que necesite de una gestión agil y sencilla para el tratamiento de la información.

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano, inglés, catalán

PROVEEDOR : IBER AMBER, S.A.

NOMBRE : RBASE 6000

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 512 KB

MEMORIA DE MASA : 1 unidad de diskette
Disco fijo : 10 MB

SIST. OPERATIVO : BTOS

SOFTWARE COMPLEMENTARIO : ISAM

FUNCION : Base de datos relacional. Multiusuario en red.

DESCRIPCION FUNCIONAL :

- Acceso relacional a las tablas
- Lógica booleana de selección
- Diseño interactivo de pantallas e informes
- Intercala con Multiplan, Lotus 1-2-3
- Protección de acceso a varios niveles

AMBITO DE APLICACION :

Grandes cuentas con necesidad de un centro de información basado en micros, que no se limiten a transferir ficheros de un mainframe que posteriormente se exploten con hojas ellectrónicas, sino que también puedan hacer consultas selectivas y se adapten los datos con la facilidad y potencia de los operadores relacionales.

MICROORDENADOR : BURROUGHS B-25

DOCUMENTACION : Castellano, inglés

PROVEEDOR : BURROUGHS, S.A.

NOMBRE : BORIAR

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB, 192 KB Disco fijo:10 MB recomendado

MEMORIA DE MASA : 1 ó 2 unidades de diskette

SIST. OPERATIVO : MS DOS 2.0 ó superior

SOFTWARE COMPLEMENTARIO :

FUNCION :

- Gestor de base de datos, generador de aplicaciones y editor.

DESCRIPCION FUNCIONAL :

- Definición de ficheros y acceso manual a los mismos
- Definición de informes y listados y ejecución individual o por lotes.
- Adaptación de ficheros a/de otras aplicaciones
- Número ilimitado de bases de datos

AMBITO DE APLICACION :

- Cualquiera de propósito general

MICROORDENADOR :

DOCUMENTACION : Castellano

PROVEEDOR : COMPUGRAF,S.A.

NOMBRE : MANAGER

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : MS-DOS/OASIS

SOFTWARE COMPLEMENTARIO :

FUNCION :

- Creación automática de ficheros. Acceso por campos que no son el índice. Apoyo con base de datos relacional. Campos ocultos.

DESCRIPCION FUNCIONAL :

- Creación de ficheros
- Listados selectivos según campos existentes

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano, inglés

PROVEEDOR : ESTUDIOS DE PROGRAMACION, S.L.

NOMBRE : GESDAT-3

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : MS-DOS

SOFTWARE COMPLEMENTARIO :

FUNCION :

- Programa de gestión de ficheros

DESCRIPCION FUNCIONAL :

- Permite la confección, entrada y rectificación de fichas, éstas pueden ser localizadas por cualquier valor de sus campos dentro del fichero.
- Las fichas entradas con el programa pueden ser ordenadas, seleccionadas o incluidas, según el valor de uno o más campos, e impresas en distintos formatos.

AMBITO DE APLICACION :

- Programa de índole general

MICROORDENADOR : EPSON QX-16 EPSON QX-11

DOCUMENTACION : Castellano

PROVEEDOR : EPSON-STI

NOMBRE : AMBER-FILE

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 1 Unidad de diskette

SIST. OPERATIVO : DOS 1.1 y posteriores

SOFTWARE COMPLEMENTARIO : Amber. Amber-graph

FUNCION : Fichero preconfigurado de referencias integrables a un sistema amber.

DESCRIPCION FUNCIONAL :

- Fichero de datos Amber tratables
- Sort automático
- Acceso indexado directo y por aproximación sobre los 50 primeros caracteres de la clave
- Ventanas de ayuda en pantalla en cualquier momento y circunstancia.

AMBITO DE APLICACION:

- Ficheros de referencias, fichero de secretaria, fichero-agenda, etc.

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano, inglés

PROVEEDOR : IBER AMBER, S.A.

NOMBRE : BASE LOGIC

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 1 Unidad de diskette

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO : Todas las aplicaciones de Logic Control

FUNCION : Libre definición y manipulación de archivos.

DESCRIPCION FUNCIONAL :

- Ficheros ordenados permanentemente por clave o código
- Forma de definición de los campos totalmente libre

- Ficheros de 60 campos y longitud de registros de 256 bytes
- Posibilidad de realizar las siguientes operaciones sobre los ficheros: actualización, listado, cálculo, eliminación, clasificación, listado etiquetas, acumulación, restauración directorio.

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC-XT/IBM PC-AT y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : LOGIC CONTROL, S.A.

NOMBRE : INFOSTART

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO :--

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

- Menús en español sin programar ni codificar. Fácil entrada de datos preparada por el propio usuario! Generador rápido de reportes. Super clasificación más rápido que cualquier otro SGBD para micros. Actualización transaccional que actualiza múltiples ficheros automáticamente, 200 combinaciones de máscaras de entrada para mayor precisión. Pantallas de ayuda con señalizaciones de página en el manual del usuario. Auto explicativo. Verificación rápida de datos en lote. Registros de longitud variable.

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : MICRONET, S.A.

NOMBRE : DATAFLEX

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 2 x 360 KB

SIST. OPERATIVO : MS-DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

Se compone de una base de datos y de un generador de aplicaciones. Permite la generación automática de sistema de menús, mantenimiento de ficheros y generador de informes. Todo esto combinado con un lenguaje de comandos potentes y flexibles, hacen prehistórica la programación en BASIC. Dataflex se suministra en 2 versiones : Módulo de desarrollo completo para el programador y run-time para usuarios. Disponibles versiones de 8 y 16 bits mono o multipuesto.

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : MICRONET, S.A.

NOMBRE : dBASE II

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 1 x 360 KB

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles // AMSTRAD CPC-464

DOCUMENTACION : Castellano

PROVEEDOR : MICRONET, S.A.

NOMBRE : INFOBASE

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : DOS 1.1 ó posterior

SOFTWARE COMPLEMENTARIO : Facturación

FUNCION : Tarjetero electrónico. Archivo de entrevistas. Archivo de proyectos.

DESCRIPCION FUNCIONAL :

- 3 programas seleccionados por menú independientes forman un mismo sistema para archivo de entrevistas, tarjetas de visita y proyectos u ofertas, permitiendo la actualización de datos en todo momento.

AMBITO DE APLICACION :

- Profesionales y todo tipo de empresas

MICROORDENADOR : IBM PC_XT y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : OFIMATICA COMERCIAL, S.A.

NOMBRE : BARBRO-P

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB Disco fijo : Opcional

MEMORIA DE MASA : 1 ó 2 x 360 KB

SIST. OPERATIVO : P-System

SOFTWARE COMPLEMENTARIO :

FUNCION : Bases de datos. Brev. Arkiv. Kartotek. Adress y relacional. Hoja electrónica y proceso de textos.

DESCRIPCION FUNCIONAL :

- Sistema modular e integrado
- Extrema facilidad y flexibilidad en la creación y manejo de ficheros
- Conversacionalidad en castellano en la mayoría de los programas

AMBITO DE APLICACION :

Centros de investigación de mercado y opinión. Departamentos comerciales y marketing de empresas. Centros universitarios y de enseñanza general.

MICROORDENADOR : IBM PC XT y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : TESI, S.A.

NOMBRE : R-BASE 4000

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 256 KB

MEMORIA DE MASA : 2 x 360 KB

SIST. OPERATIVO : MS/DOS, CTOS, BTOS

SOFTWARE COMPLEMENTARIO : ---

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : RHV IBERICA, S.A.

NOMBRE : DATA-BASE

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 KB Disco fijo : 10 MB

MEMORIA DE MASA : 2 x 360 KB

SIST. OPERATIVO : MS/DOS

SOFTWARE COMPLEMENTARIO : --

FUNCION : Gestión de ficheros y generación de informes

DESCRIPCION FUNCIONAL :

- Diseño y mantenimiento de ficheros. Consultas y listados clasificados automáticos. Acceso a ficheros en base a otros existentes. Soporte de varios niveles de índices.

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Castellano

PROVEEDOR : ULDATA S.COOP

NOMBRE : FIRST-CLASS MAIL

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 128 K RAM

MEMORIA DE MASA : 1 ó 2 unidades de diskette

SIST. OPERATIVO : DOS 1.1

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos útil para la elaboración de mailings

DESCRIPCION FUNCIONAL :

- Programa para generar etiquetas y listados de un mailing.
- Permite la ordenación por nombre, Compañía u otro criterio que se escoja.
- Está diseñado para utilizarlo con la opción Mailmerge del Wordstar ó otros procesadores de texto que tengan una opción similar al Mailmerge.

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : CONTINENTAL SOFTWARE

NOMBRE : TIM III

REQUERIMIENTOS DEL SISTEMA:

MEMORIA MINIMA : 128 K mínimo

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : DOS 1.1

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

- Programa manejable mediante menús.
- Permite: añadir, analizar, actualizar registros, crear ficheros, generar listados e informes. Es de fácil manejo.

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : INNOVATIVE SOFTWARE

NOMBRE : VISIFILE

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 64 K mínimo

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : DOS 1.1 ó 2.0

SOFTWARE COMPLEMENTARIO :

FUNCION :

DESCRIPCION FUNCIONAL :

- Programa orientado a formas. Permite crearlas, introducir información en ellas y recopilar información en función de diferentes criterios. Permite también imprimir datos en informes sencillos. A pesar de que pretende ser bastante flexible, es fácil perderse con los menús. No permite realizar ninguna copia del programa. Visicorp tampoco suministra una copia de seguridad al comprarlo, pero vende una cuando se devuelve la tarjeta de garantía.

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : VISICORP

NOMBRE : CONDOR

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 80 K mínimo

MEMORIA DE MASA : 2 unidades de diskette

SIST. OPERATIVO : DOS 1.0 ó 1.1

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de daos orientada fundamentalmente a cuentas

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : CONDOR 3

NOMBRE : PFS:FILE y PFS:REPORT

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 64 K

MEMORIA DE MASA : 1 ó 2 unidades de diskette

SIST. OPERATIVO : DOS 1.0, 1.1 ó 2.0

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION :

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION :

PROVEEDOR : SOFTWARE PUBLISHING CORPORATION

NOMBRE : REVELATION

REQUERIMIENTOS DEL SISTEMA

MEMORIA MINIMA : 320 K

MEMORIA DE MASA :

SIST. OPERATIVO : DOS y MS/DOS (Lleva un sistema operativo propio: PICK compatible con los comandos visuales de los anteriores).

SOFTWARE COMPLEMENTARIO :

FUNCION : Base de datos que consta de: generador de pantallas, editor y lenguaje de creación de procedimientos para clasificaciones y extracciones de registros multiclaves.

DESCRIPCION FUNCIONAL :

- Ver descripción detallada

AMBITO DE APLICACION :

- General

MICROORDENADOR : IBM PC y compatibles

DOCUMENTACION : Inglés

PROVEEDOR :

SISTEMAS DE INFORMACION DOCUMENTAL 5



L desarrollo de las posibilidades de los ordenadores personales ha permitido el desarrollo de programas para el almacenamiento y recuperación de información documental capaces de aprovechar eficazmente las posibilidades de los microordenadores profesionales. Ello está haciendo posible que muchas bibliotecas, centros de documentación y unidades de información en general puedan mecanizar la gestión de sus procedimientos de trabajo y abordar la creación de bases de datos documentales automatizadas.

Entenderemos aquí por información documental cualquier tipo de información textual, físicamente estructurada bajo la forma de monografías, artículos de revista, dossieres, informes, memorias, etc., o bajo la forma de descripciones de «objetos» que no son documentos: historiales clínicos, biografías, información descriptiva de instituciones y entidades de monumentos, investigaciones en curso, especificaciones de contratos o de trámites administrativos, etc...

Hasta el año 1982 y debido sobre todo a la escasez de programas específicos, no se había generalizado todavía el empleo de microordenadores en centros y servicios de documentación, bibliotecas y unidades de información en general. Los escasos paquetes de Software existentes estaban diseñados para ordenadores de 8 bits con el sistema operativo CP/M, que en muy pocos casos contaban con unidades de disco duro para almacenamiento masivo de datos. A partir del año 1983, sin embargo, con la introducción masiva del PC IBM en el mercado, y particularmente de su versión profesional, el PC/XT equipado con un disco duro de 10 Mb, la situación cambia de forma notable y la oferta de programas especializados en la gestión de información documental aumenta sustancialmente. Con la transformación del PC en el estándar del mercado de ordenadores, se asiste al desarrollo de nuevos programas que buscan aprovechar al máximo

sus capacidades (mayor velocidad, discos duros fijos de gran capacidad, etcétera), a la conversión de las antiguas versiones de programas desarrollados para el sistema operativo CP/M al sistema operativo MS-DOS, y al desarrollo de versiones para MS-DOS de paquetes de software de recuperación de información y gestión documental que hasta ese momento sólo podrían ejecutarse en grandes ordenadores y/o en miniordenadores. Estas nuevas tendencias se manifiestan sobre todo en los Estados Unidos y en algunos países europeos: Gran Bretaña, Francia, Alemania y países escandinavos (nuestro país ha quedado al margen de las mismas y sólo a finales de 1984 y a lo largo del año 1985 algunas firmas españolas han comenzado a desarrollar y comercializar programas originales o a adquirir los derechos de distribución de paquetes de software extranjeros).

El desarrollo de los «super-micros» —el PC AT y compatibles— de la tecnología de redes locales y de los microordenadores multiusuario basados en implementaciones del sistema operativo MS-DOS o en el sistema operativo UNIX (XENIX, PC-IX) ha ampliado el campo de aplicación de los microordenadores en tareas de gestión de información documental a los entornos personal, profesional o de las pequeñas empresas e instituciones a los entornos propios de las medianas e incluso grandes organizaciones.

Dada la gran diversidad de productos que hay en el mercado es preciso, para una mecanización con éxito, tener en cuenta en primer lugar los distintos tipos de programas desarrollados para aplicaciones documentales:

A) Sistemas de gestión documental.

B) Sistemas de gestión de bibliotecas.

C) Programas «gateway» para comunicación con las grandes bases de datos comerciales.

D) Programas para la recuperación de información en ficheros generados por programas de tratamiento de textos.

E) Programas para la automatización de las tareas de desarrollo de herramientas documentales.

A continuación analizaremos con más detalle cada uno de estos tipos.



SISTEMAS DE GESTION DOCUMENTAL

Especialmente concebidos para gestionar bases de datos con información textual. Los usuarios están en condiciones de refinar sus preguntas hasta obtener de la base de datos, combinando criterios de búsqueda múltiples y con una gran rapidez en la recuperación, la información que necesitan.

Este tipo de programas permiten desarrollar aplicaciones parametrizables adaptables a necesidades de gestión de información muy diversas con una única o con ninguna programación suplementaria.

En nuestro país, dados los bajos niveles de utilización y conocimiento de estos programas, suele recurrirse a los sistemas de gestión de bases de datos (SGBD) para sustituirlos, intentando así satisfacer de forma muy precaria necesidades de gestión de información documental.

Los SGBD no son capaces de administrar ningún tipo de base de datos documental, ya que están concebidos para gestionar bases de datos factuales o numéricas, respondiendo eficazmente tan sólo si los usuarios plantean preguntas muy precisas dentro de un repertorio limitado de posibilidades de interrogación.

Las características básicas generales de un paquete de software de gestión son:

— Los SGD son programas concebidos para almacenar y recuperar fácilmente información de tipo textual.

— Los SGD almacenan registros descriptivos de documentos, divididos en varias zonas (campos o parágrafos) que contienen información específica.

— Los SGD son capaces de recuperar con gran rapidez todos los registros que satisfacen los criterios contenidos en una ecuación de búsqueda mediante la utilización de la técnica de ficheros invertidos.

— Los SGD permiten interrogar una base de datos combinando lógicamente (mediante operadores booleanos: AND, OR, NOT) múltiples criterios de búsqueda.

— Muchos SGD incluyen módulos específicos para la utilización y gestión de thesauros, vocabularios controlados que permiten controlar las relaciones semánticas entre los términos que contienen y eliminar las ambigüedades semánticas durante las fases de descripción (indización).

En resumen, las tareas básicas que es capaz de ejecutar un SGD son:

1. Creación y gestión de bases de datos: entrada, corrección y eliminación de registros.
2. Búsquedas de información en línea con tiempos de respuesta muy cortos utilizando lógica Booleana.
3. Edición en pantalla o en impresora de los resultados de las búsquedas y producción de catálogos e índices impresos.

Otras funciones características de los SGD son: procedimientos de mantenimiento que recuerdan al usuario que debe crear copias de seguridad de las bases de datos; protección del contenido de las mismas mediante el establecimiento de niveles de acceso («passwords» para las diferentes categorías de usuarios); facilidades de interrogación asistida para los usuarios no familiarizados con el sistema; software multiusuario; almacena-

miento de las estrategias de búsqueda para que puedan ser reutilizadas en cualquier momento; posibilidad de formatear las salidas impresas (encabezamientos, márgenes, paginación, etc).

El campo de aplicación de este tipo de programas es muy amplio: creación local de bases de datos bibliográficos (con referencias de todo tipo de documentos: libros, artículos de revistas, informes, memorias, proyectos, etc.); bases de datos con información sobre recursos de personal, instituciones, historias clínicas, catálogos de productos; bases de datos para control de ventas, seguimiento de clientes, control de correspondencia, gestión de suscripciones, producción de anuarios, catálogos, directorios, etcétera.

Los tipos de usuarios que pueden obtener un gran rendimiento de un sistema informático integrado por un PC y un paquete de Software documental son: Centros de Información y Documentación, servicios de documentación de empresas, centros de investigación, firmas consultoras, gabinetes y despachos profesionales, medios de comunicación, Ayuntamientos, editoriales, laboratorios, departamentos de Administración pública, Universidades, Hospitales, etc.

La mayor parte de los paquetes SGD para microordenadores han sido desarrollados y comercializados en Estados Unidos. En Europa, tan solo en Francia y sobre todo en Inglaterra existen algunos desarrollos y comercialización de programas SGD. Por lo que se refiere a nuestro país, se han desarrollado muy pocos productos y muchos no han sido todavía —o no lo serán— lanzados al mercado.



SISTEMAS DE GESTION DE BIBLIOTECAS.

Omitimos aquí su discusión por haber sido desarrolladas anteriormente en este libro.

```
LMS-83 LIBRARY MANAGEMENT SYSTEM
  *- D E M O   L I B R A R Y -*
        * MAIN MENU *
  1.    ITEM ENTRY & UPDATE
  2.    ITEM ENQUIRIES
  3.    READERS' ENTRY & LENDINGS
  4.    REPORTS
  5.    CIRCULATION LISTS
  6.    KEYWORDS & THESAURUS
  7.    MAINTENANCE PROCEDURES
  99    EXIT
PLEASE ENTER REQUIRED FUNCTION
NUMBER                  PASSWORD
```

Fig. 5.1. Menú principal del sistema Libby.



Programas «gateway» para la conexión e interrogación asistidas de grandes bases de datos comerciales

Durante los cinco últimos años se ha producido un impresionante desarrollo de los servicios comerciales facilitados por los grandes distribuidores de bases de datos (información científica y técnica, económica, comercial, estadística, etc.) para soportar la interrogación en línea por parte de los usuarios de microordenadores. Dos de estos distribuidores, DIALOG y BRS Information Technologics, han puesto en marcha servicios de acceso (Knowledge Index de DIALOG y BRRTHRV de BRS) a sus bases de datos, que utilizan lenguajes de interrogación simplificados y son facturados con tarifas reducidas.

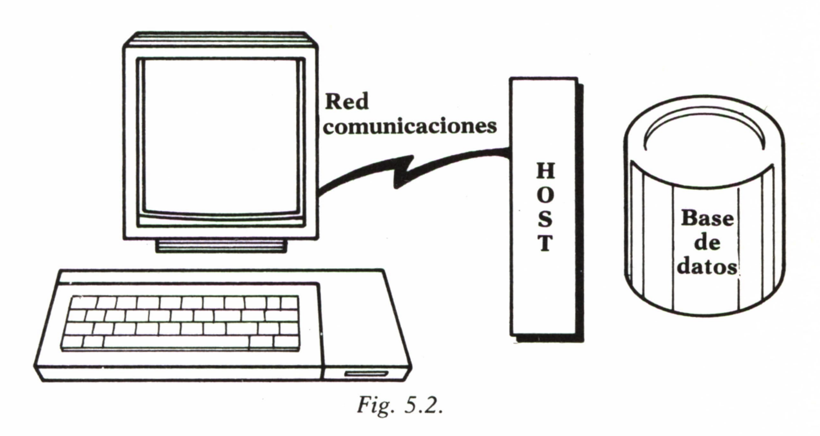


Fig. 5.2.

Como resultado, o acompañando a la oferta de estas facilidades, los propios distribuidores y algunas empresas estadounidenses han desarrollado los paquetes de software «gateway», potentes programas de comunicaciones que incorporan funcionalidades de interrogación. Estos programas son capaces, al igual que muchos otros programas de comunicaciones, de intercambiar datos con grandes ordenadores remotos, pero lo que les diferencia de aquéllos es que constan de elaboradas interfaces que permiten que las grandes bases de datos comerciales puedan ser consultadas por usuarios que desconocen sus, a veces, complicados lenguajes de interrogación. Los paquetes de software «gateway» simplifican esta tarea al máximo reduciéndola a la introducción de una palabra o una frase en el terminal.

La utilización de los programas «gateway» no le evita, sin embargo, al usuario el tener que aprender los fundamentos de la estrategia de la búsqueda en línea (localización de los términos de búsqueda correctos, elaboración de la ecuación de búsqueda, análisis de sus elementos, etc.).

Estos programas han sido diseñados para aprovechar al máximo las capacidades de proceso y comunicaciones del PC y simplificar los procedimientos de conexión e interrogación reduciéndose los costes de utilización del servicio (tarifas de utilización de las redes telefónicas o de transmisión de datos y tarifas por tiempo de conexión del distribuidor).

Las ventajas que proporcionan los sistemas formados por un PC y un paquete de Software «gateway» son tales que progresivamente están reemplazando a los terminales normales. Algunas de estas ventajas son:

a) Procedimientos simplificados para establecer la conexión con el ordenador «host» (logging-on»): una forma de simplificar la conexión consiste en almacenar de forma permanente los «passwords» y los identificadores de red y de «host» del usuario, con lo que el proceso de establecer una conexión queda reducido a unas cuantas pulsaciones del teclado.

b) Elaboración «off-line» de la ecuación de búsqueda: Si elaboramos la ecuación de búsqueda antes de establecer la conexión con la base de datos, ahorramos los costes del mayor tiempo de conexión que supondría elaborarla en línea. Después de preparar y comprobar la ecuación de búsqueda, ésta es enviada al ordenador «host» pulsando una tecla de función que antes ha establecido la conexión y seleccionado la base de datos. Las ecuaciones de búsqueda que se utilizan frecuentemente pueden ser almacenadas para ahorrar tiempo de tecleado y para crear perfiles de búsqueda que permita una difusión selectiva de la información.

c) «Downloading» (proceso de capturar datos de un ordenador «host» para almacenarlos bajo una forma legible por la máquina): los resultados de la búsqueda pueden ser volcados y almacenados en las unidades de disco del microordenador; esto hace posible que:

1. Los resultados de la búsqueda puedan ser editados, eliminando referencias no pertinentes, duplicando otras o elaborando un informe con una buena presentación con ayuda de un programa de tratamiento de textos.

2. Los resultados de la búsqueda pueden ser transferidos al sistema de información local soportado por el propio microordenador, utilizando un programa de «reformateo» que adapte el formato de los datos recuperados al utilizado por el sistema propio.

Algunos paquetes «gateway» pueden ser integrados por programas de gestión documental que simplifica este proceso y que permiten crear bases de datos locales, que hacen posible refinar las búsquedas, añadir información propia, etc.

d) Almacenamiento de información de ayuda (información sobre órdenes de los lenguajes de interrogación) que puede ser consultada durante las búsquedas.

Características generales de los paquetes «gateway» disponibles en el mercado norteamericano

— Requerimientos de 128 Kb de memoria RAM y unidad de disco.

— Velocidad de transmisión de 300 a 1.200 bps.

— Soporte para el acceso en línea a los distribuidores DIALOG, ORBIT y BRS.

— Utilidades: conexión y marcado automático, elaboración off-line de la ecuación de búsqueda, «down-loading» y selección de la base de datos.

Algunos paquetes «gateway» que se comercializan actualmente:

Estados Unidos: SCI-MATE UNIVERSAL ONLINE SEARCHER, PC/NET-LINK, SEARCH MASTER, INSEARCH, SEARCH HELPER, SEARCH-WARE.

Europa: El distribuidor ESA/IRS alquila a sus usuarios el paquete MAGPIE junto con el microordenador Columbia.

España: FUINCA (Fundación de la Red de Información Científica Automatizada) está desarrollando un paquete «gateway» para soportar las comunicaciones con las principales bases de datos españolas.



Programas para la automatización de las tareas de desarrollo de herramientas documentales

El trabajo de documentación ha de estar soportado por ciertas herramientas o instrumentos técnicos. El desarrollo de muchos de estos instrumentos puede ser considerablemente abreviado y mejorado, recurriendo a funciones específicas de los sistemas de gestión documental (SGD) o a paquetes de software especializado. Así, ya es posible automatizar la construcción de índices (técnicas KWIC, KWOC, KWAC y NEPHIS), la creación y mantenimiento de thesauros (paquetes ASSAIN, MICRO-CAIRS, etcétera).

El uso de microordenadores para el tratamiento y la gestión documentales se está desarrollando rápidamente. Sin embargo, su empleo origina también algunos problemas que pueden ser resueltos si se elabora una buena documentación del sistema, se implantan procedimientos normalizados para la entrada de datos y la utilización del programa y se proporciona una formación suficiente al personal a cargo del sistema.



FICHAS TECNICAS



Aquila

SISTEMA DE RECUPERACION DE INFORMACION TEXTUAL EN LENGUAJE LIBRE

AQUILA soporta todos los procesos necesarios para crear, interrogar, mantener e imprimir bases de datos documentales. Utiliza una arquitectura de ficheros inversos, de eficacia probada para recuperar información de bases de datos en lenguaje libre.

BASE DE DATOS

— N. de bases de datos : Ilimitado.
— N. de registros : Sólo limitado por el sistema.
— Longitud del registro: 2.250 caracteres.
— N. de campos : 100.
— Facilidades de edición: Caracteres aislados, palabra por palabra y edición de línea.
— Etiquetado de campo : Etiquetas de identificación y «prompts» para cada campo definido por el usuario.
— Lista de palabras vacías: Dispone de una lista predefinida de 80 palabras; el usuario puede crear su propia lista.

SISTEMA DE INVERSION

Generación de índices por inversión: Totalmente automática con compactación de datos para optimizar ocupación de disco.
Estadísticas: Facilita datos sobre el número de términos indicados y clasificados.
N. de ocurrencias de los términos: Facilita listados con datos sobre el número de veces que aparece un término en el índice y sobre el tipo de campo en el que aparece.

SISTEMA DE BUSQUEDA

Lenguajes de interrogación: Basado en el CEC. Common Comand Language (EURONET), utiliza órdenes como FIND, SHOW, PRINT.

Operadores lógicos: AND, OR, NOT.

Truncado: Trunca a la derecha.

Almacenado de la ecuación de búsqueda: Las ecuaciones de búsqueda pueden ser grabadas y reutilizadas en cualquier momento.

Historia de la búsqueda: La orden SHOWSEARCH permite consultar en pantalla la estrategia de búsqueda empleada.

SALIDAS

— Selección de campos específicos para su impresión.

— Determinación del orden en el que los campos seleccionados deberán ser impresos.

— Capacidad de utilizar formatos de impresión predefinidos por el usuario.

— Capacidad de almacenar hasta 10 formatos de impresión por cada base de datos.

— Capacidad para generar nuevos formatos de impresión o modificar los existentes.

— Capacidad para imprimir en negrilla, subrayar o sangrar el contenido de campos específicos.

HARDWARE

AQUILA funciona en el IBM PC y compatibles, SIRIUS, WANG, DEC RAINBOW y TELEVIDEO.

PRECIO: 98.000 pesetas.

DISTRIBUCION: I+D Informática y Documentación, S. A. Teléf. 411 02 33.



Docutex

SISTEMA DE ALMACENAMIENTO DE INFORMACION

Docutex almacena la información en forma de fichas; un pequeño editor incorporado permite editar fichas hasta una longitud de 512 caracteres (aproximadamente 80 palabras). Almacena y numera automáticamente las fichas a medida que se van creando: es posible asignar a cada ficha palabras clave o referencias que serán utilizadas en el momento de recuperar la información mediante operadores lógicos Booleanos. Estas referencias pasan a formar parte de un diccionario que se mantiene ordenado alfabéticamente y que puede ser consultado en línea en el momento de la

entrada de datos y al interrogar el fichero para recuperar información. Es posible también imprimir las fichas recuperadas en formatos que define el usuario, o crear ficheros de disco que pueden ser editados con un programa de tratamiento de textos.

CARACTERISTICAS TECNICAS

— Número de ficheros en disco: Ilimitado.
— Longitud de la ficha : 512 caracteres.
— Diccionario : Hasta 32.500 caracteres.
— Referencias : Hasta 32.500 referencias distintas por ficha.
— Búsquedas: Por palabras exactas, aproximadas o genéricas.
— Operadores lógicos : , U, —, sobre conjuntos.
— Comandos: : B-Borrar, M-Modificar, D-Diccionario, E-Editor, V-Visualizar, L-listar, etc.

HARDWARE

DOCUTEX funciona en IBM PC y compatibles. Requiere 128 Kb y 2 unidades de disco.

DISTRIBUCION:

MICRONET, S. A. Víctor de la Serna, 36.
28016 MADRID. Tel. 457 50 56



TEXTO

SISTEMA DE GESTION DOCUMENTAL UNIVERSAL

Diseñado originalmente (1976) para grandes volúmenes de información en grandes sistemas. La versión para PC aparece en 1985 y es orientado a usuarios sin conocimientos de informática.

CATALOGO

Permite describir y memorizar los parámetros de las distintas aplicaciones desarrolladas por el usuario. Los parámetros de una aplicación pueden modificarse en cualquier momento.

Descripción de la aplicación documental: La parametrización permite definir:

— Los ficheros documentales: características de la numeración de los documentos, lista de nombres de los campos, tipo de campos (numéricos, implícitos en la interrogación, no modificables en su presentación).

— Los índices asociados.

— Los ficheros relacionados.

Descripción de las ediciones: Los documentos de edición o de tabulación permiten elaborar presentaciones complejas y bien estructuradas de los documentos para revalorizar la información.

El usuario puede: editar sólo lo que es útil; presentar los distintos campos, en el orden que se desee, unos encima de otros o en presentaciones en columnas (tablas); introducir encabezamientos y subencabezamientos; numerar las páginas; insertar códigos de fotocomposición para preparar ficheros que puedan ser enviados a la imprenta, etc.

Diálogos: Permiten diseñar procedimientos para lanzar una serie de órdenes que pueden ir acompañadas de comentarios, ahorrando al usuario la ejecución de secuencias repetitivas y tediosas.

FICHEROS DOCUMENTALES

— N. de ficheros: Ilimitado en el espacio de disco disponible.

— N. de registros : Ilimitado en el espacio de disco disponible.

— Formato de las fichas: Variable (*).

— N. de campos: 99 (*).

— Formato de los campos: Variable.

— Campos interrogables: Todos (sin embargo, en el caso de aplicaciones con un gran volumen de registros, los campos deben invertirse en los índices).

— Modificaciones: Prácticamente todas son posibles.

INDICES

Cuando los campos interrogables son invertidos en los índices, los tiempos de respuesta son siempre instantáneos.

— N. de registros: Ilimitado.

— N. de índices: Un fichero puede estar relacionado con varios índices (*).

— Inversiones acumuladas: Varios campos pueden estar invertidos en un mismo índice.

— Puesta al día: Por lotes o en tiempo real.

(*) LIMITES «STANDARD»: 4.000 caracteres y 99 campos por registro, 10 índices por fichero. Estos límites pueden modificarse en la medida que el sistema de la máquina utilizada lo permita.

LENGUAJE DE ORDENES

Las funciones básicas de TEXTO son poco numerosas y fáciles de aprender. Las órdenes básicas pueden ser combinadas entre sí para ejecutar tareas más complejas.

HARDWARE

TEXTO funciona en el IBM PC/XT y compatibles y en el IBM/PC/AT y compatibles. Sistema operativo MS-DOS.

Requiere 512 Kb de memoria y disco duro.

PRECIO: 365.000 pesetas

DISTRIBUCION:

I+D Informática y Documentación, S. A. Teléf. 411 02 33



Infos

SISTEMA DE CONTROL DE BASES DE DATOS DE INFORMACION DOCUMENTAL INFOS

El Sistema INFOS está dividido en 4 grandes módulos, cada uno de ellos con diferentes funciones, según se muestra en la fig. 5.3.

DIALOGOS

Todas las funciones se ejecutan mediante un menú de interfase con el usuario.

FICHEROS DOCUMENTALES

— N. de registros: Ilimitado.
— N. de campos: Ilimitado.
— N. de bases de datos: Hasta 32 bases de información. Cada una con un formato predefinido por el usuario.
— Edición: Editor de textos especializado en masas de información.

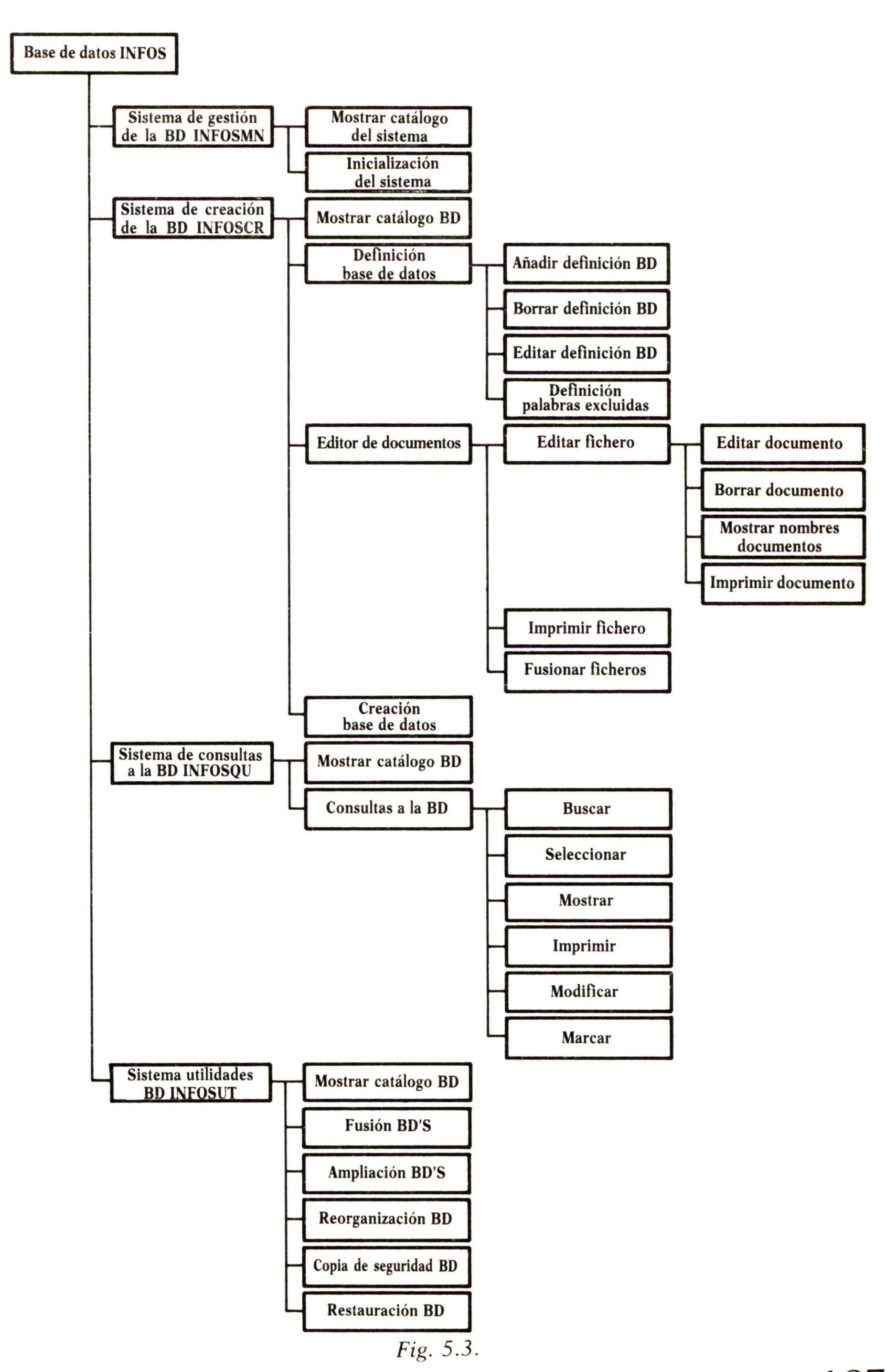

Base de datos INFOS
Sistema de gestión de la BD INFOSMN
Mostrar catálogo del sistema
Inicialización del sistema
Sistema de creación de la BD INFOSCR
Mostrar catálogo BD
Definición base de datos
Añadir definición BD
Borrar definición BD
Editar definición BD
Definición palabras excluidas
Editor de documentos
Editar fichero
Editar documento
Borrar documento
Mostrar nombres documentos
Imprimir documento
Imprimir fichero
Fusionar ficheros
Creación base de datos
Sistema de consultas a la BD INFOSQU
Mostrar catálogo BD
Consultas a la BD
Buscar
Seleccionar
Mostrar
Imprimir
Modificar
Marcar
Sistema utilidades BD INFOSUT
Mostrar catálogo BD
Fusión BD'S
Ampliación BD'S
Reorganización BD
Copia de seguridad BD
Restauración BD

Fig. 5.3.

CATALOGO DEL SISTEMA

Realiza las siguientes funciones:

— Función de búsqueda: (SEARCH) con la ayuda de operadores Booleanos, el sistema puede ejecutar la búsqueda de palabras sueltas o de frases completas. La búsqueda se efectúa sobre los niveles siguientes: documento, párrafo, oración, partes de la oración.

— Función de selección: (SELECT) Hace referencias selectivas desde los documentos seleccionados previamente con la función de búsqueda.

— Función de consulta: Permite al usuario mostrar en la pantalla del terminal todos los documentos pertenecientes a la lista resultante de una consulta determinada.

Existen muchas otras funciones: IMPRIMIR, MODIFICAR...

HARDWARE

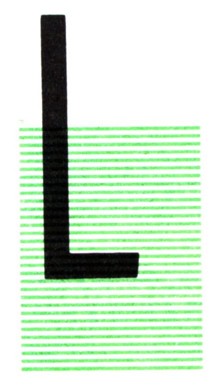
INFOS funciona con IBM PC/XT y compatibles.

Requiere 256 Kb y disco duro (10 Mb).

Funciona con MS-DOS y se espera una versión para UNIX en el futuro, pues está desarrollado en lenguaje «C».

DISTRIBUCION: SPS IBERICA, S. A.

COMO ELEGIR UNA BASE DE DATOS 6



LA elección de una base de datos es un tema complejo y difícil. En general no se puede decir que una base de datos sea mejor que otra. Cada una es buena para alguna aplicación. Así una tendrá mayor velocidad en búsqueda y ordenamiento, pero no permitirá incluir campos de seguridad, mientras que otra podrá incluir estos campos pero trabajará con un lenguaje que será de difícil comprensión para el usuario no especializado.

Por tanto, habrá que sopesar muchos factores para elegir definitivamente la que consideremos adecuada.

No toda base de datos puede sernos útil para el problema que vayamos a abordar.

Aunque pueda parecer una perogrullada, lo primero que habremos de hacer es plantearnos muy en detalle para qué aplicaciones la vamos a usar y no al revés, esto es, primero comprar la base de datos y luego amoldarnos a sus limitaciones y capacidades. Si alguien se tiene que amoldar es la base de datos a nosotros, no nosotros a ella.

En el mercado existen las suficientes bases de datos como para que alguna se amolde a nuestras necesidades.

Entre las que considere que le pueden ser útiles no compre la más barata. En informática, lo más barato no es siempre lo mejor, ni lo más apropiado.

A continuación veremos algunos puntos que son importantes a la hora de elegir nuestra base de datos.

Algunos puntos son esenciales (como la compatibilidad con el ordenador) pero otros no. La importancia de estos está en función de las necesidades a las que sea orientada.



REQUISITOS HARDWARE

Suponiendo que hayamos analizado en profundidad las aplicaciones para las que vamos a utilizar la base de datos, lo segundo es averiguar las características de nuestro ordenador (lo hallaremos en la ficha técnica que nos proporciona el fabricante).

Nuestro ordenador poseerá unas determinadas características que van a limitar la elección que hagamos.

Hay que tener cuidado: no todas las bases de datos van a ser compatibles con vuestro ordenador, esto es, nuestro ordenador puede que no reúna todos los requisitos de hardware especificados para que la base de datos funcione bien.

Estos requisitos hardware son:

— Capacidad mínima de memoria que posea nuestro equipo: las bases de datos necesitan cierta cantidad de memoria para que puedan trabajar correctamente.

— Sistema operativo de nuestro ordenador. Cada base de datos especifica un tipo de sistema operativo. Generalmente éste no es único y pueden acoplarse bien a varios.

— Modelo de ordenador.

— A veces también se especifica el microprocesador que lleva el ordenador.

Hay que recordar que en cuanto nuestro ordenador no reúna cualquiera de los requisitos hardware especificados por el fabricante de la base de datos, no nos servirá.

Dada la amplia gama de bases de datos que hay en el mercado, no tendremos problemas en elegir la que más nos convenga.

En el capítulo se realiza un estudio de algunas de las principales bases de datos, explicitándose aquellos requisitos hardware necesarios.



CAPACIDAD DE ALMACENAMIENTO. ESTRUCTURA

Una vez que hayamos pensado qué tipo de aplicación queremos hacer y qué bases de datos son compatibles con nuestro ordenador, debemos realizar un esquema de cómo vamos a estructurar nuestra base de datos.

¿En qué consiste hacer la estructura de una base de datos? Para explicarlo, tendremos que volver al tema de ficheros, registros y campos.

Estructurar una base de datos es definir el tamaño de registros, longitud de campo y todo aquello que tiene que ver con las dimensiones que va a ocupar la información.

Si estamos organizando nuestra agenda con la distribución de campos que muestra la figura 1, nos interesa estructurar nuestra base de datos con pocos campos y muchos registros (suponiendo que vamos a introducir información referente a una cantidad grande de personas).

Por el contrario, si estamos organizando la nómina de una empresa, necesitaremos, como se muestra en la figura 6.1, un número respetable de campos.

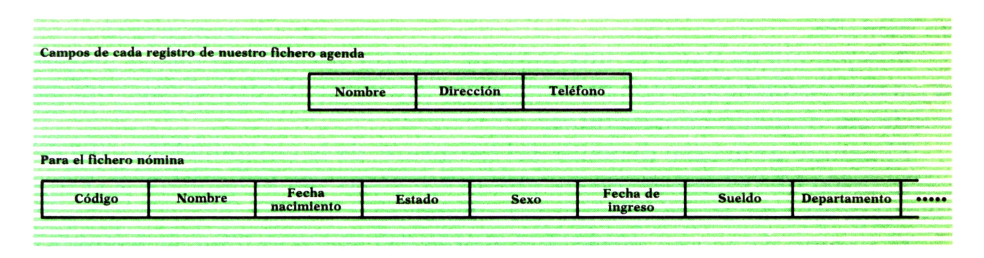


Fig. 6.1.

Cada base de datos posee unas características distintas que nos facilitarán o limitarán la estructura que de ella queramos hacer.

Las características según las cuales nos debemos guiar son:

— Número máximo de campos por registros.
— Número máximo de registros por fichero.
— Longitud máxima de cada campo.
— Tipos de datos que acepta (numérico, alfanumérico).

En el capítulo viene un estudio de algunas bases de datos, especificándose estas características para cada una de ellas.

Conviene no confundir la estructura de la base de datos con la capacidad de la misma.

La capacidad no depende del tipo de base de datos que elijamos, sino del tamaño (capacidad) del disco con el que trabajemos. Por supuesto, cuanta más capacidad tenga el disco (rígido o flexible) más caro será.

Ya que el dinero es casi siempre el que impone más limitaciones, habrá que evaluar cuanta información queremos almacenar con el fin de adquirir un disco de uno u otro tamaño.

La capacidad de los discos se mide en Kbytes. Si no va a necesitar grandes cantidades de memoria, con uno de 256 Kb le valdrá. Con uno de 1 Mb (Megabytes) tendrá suficiente espacio para cualquier aplicación.



VELOCIDAD

Para medir cuán veloz es una base de datos se usan dos parámetros:

1) TIEMPO DE ACCESO

Es el tiempo que transcurre entre una petición de dato y su aparición en pantalla.

Aunque pueda parecer una tontería esperar un minuto más o menos, si estamos accediendo constantemente a datos, ese tiempo al cabo del día puede ser bastante considerable.

El tiempo de acceso depende del hardware, de la computadora y del soporte físico que elijamos (disco, cinta). Un disco rígido tiene un acceso a los datos mucho mayor que un disco flexible.

El tiempo de acceso también depende del modo en que accedamos a la información (directo, secuencial, indexado).

El hacer comparaciones entre acceso no es fiable si no se conoce el método por el que se recuperan los datos. Así el secuencial será más rápido en cualquier base de datos que el indexado más veloz que podamos encontrar.

Si el acceso a datos, o a manipulaciones en las que es necesario acceder a ellos, son continuas y frecuentes en la aplicación que vayamos a hacer, convendrá buscar una base de datos con tiempos de acceso rápidos (podemos incluso soportar la información con un disco rígido para ganar velocidad).

Si, por el contrario, nuestra aplicación no va a ser dependiente de este tiempo, podremos ahorrarnos los gastos de disminuir el tiempo de acceso.

El otro parámetro es la:

2) VELOCIDAD DE PROCESAMIENTO

Mide la cantidad de registros que se pueden procesar por minuto.

Las pruebas que se hacen para medir este tiempo son:

— La indexación: buscar un elemento por medio de un campo-clave.

— Clasificación de los registros según el valor que tenga un campo-clave.

Por ejemplo, si el campo-clave es la edad de los obreros de una fábrica y quiero ordenarlos de más viejo a más joven.

Siempre será mejor la base de datos cuantos más registros procese por minuto, pero no debemos concederle mayor importancia si el tiempo no es definitivo para nuestra aplicación.



COMUNICACION CON LA BASE DE DATOS

Un tema de capital importancia a la hora de comprender y poder interactuar con la base de datos es el modo según el cual esté controlada.

Una base de datos puede estar controlada por:

— Menú
— Ordenes
— Programas

A) La base de datos controlada por menú es la más sencilla de usar, ya que no se necesita tener más conocimientos de informática que los estrictamente básicos.

Por supuesto, hay menús más legibles que otros; por eso a veces el tratamiento con una base de datos puede ser más dificultoso por menús mal explicados que por órdenes fácilmente aplicables.

En general, el fabricante de base de datos procura que sus menús sean claros; ahora todos vienen en español, con lo que se facilita mucho la comprensión.

B) Las órdenes son instrucciones que nos permiten acceder, modificar y realizar otras operaciones sobre cualquier parte de la base de datos.

En el dBASE II si queremos ver el contenido de un fichero teclearemos:

DISPLAY ALL

analizando de ese modo los campos de todos los registros del fichero.

Si quisiera ver sólo el campo «precio» haga:

DISPLAY ALL PRECIO OFF

Con estos ejemplos apreciamos que, aun no siendo muy complicado su uso, requerirá un estudio previo de las órdenes que acepta la base de datos.

Hemos de averiguar si lo que proyectamos comprar posee órdenes claras o, por el contrario, el conjunto de órdenes es difícil de interpretar y aplicar.

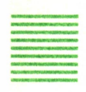
No pensemos que los Manuales que nos entreguen al comprar la base de datos van a resultar útiles para resolver cualquier duda. A veces no son todo lo claros que debieran.

C) Algunas bases de datos ofrecen la posibilidad de implementar programas con ellas.

Puede ser una gran ventaja en aplicaciones complejas.

El único inconveniente es que para saber programar hay que conocer el lenguaje, y esto no es una tarea sencilla. Suelen usar lenguajes especiales. A diferencia de los lenguajes clásicos como el BASIC, COBOL, los len-

guajes de las bases de datos no están restringidos por normas caducas e introducen sus propias órdenes y las estructuras que mejor se acoplan a la base de datos para la que sirven. La conexión del diccionario de datos con el lenguaje permite introducir campos del registro por el nombre sin tener que declararlos en el programa.



FICHEROS MULTIPLES

Supongamos que utilizamos una base de datos que sólo puede manejar un fichero a la vez, por ejemplo, la lista de clientes o la lista de productos.

Si quisiéramos sacar un informe de clientes y productos juntos no podríamos hacerlos en una base de datos que trabaje con un solo fichero.

Como alternativa surgen las bases de datos multifichero que permiten procesar dos o más ficheros a la vez.



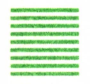
SISTEMAS MULTIUSUARIO

Es importante saber el número de personas que van a trabajar con la base de datos simultáneamente.

Si estamos trabajando sobre nuestra agenda, seguro que no nos hace falta un sistema multiusuario.

Una base de datos multiusuario requiere disponer de un equipo con varias pantallas y un sistema operativo que esté capacitado para ello.

Si, por el contrario, el Ministerio de Hacienda está recopilando información sobre las Declaraciones de la Renta de todo los españoles, necesitará un sistema multiusuario, ya que una persona sola no podría realizar todo el trabajo.



TRANSFERENCIA DE INFORMACION

En este apartado nos referiremos a la capacidad de una base de datos de poder recibir información de otros programas y de otras bases de datos.

A veces no es posible leer o escribir datos en la base de datos con programas escritos en BASIC, COBOL y otros lenguajes.

La potencia de una base de datos determinará el que esto se pueda conseguir mediante algún «truco».

En cuanto a leer datos desde otras bases puede ser importante en sistemas que manejen mucha información y necesiten estar en contacto con otras fuentes para actualizarse.



SEGURIDAD

Por seguridad entendemos la capacidad que posee la base de datos para que los datos que contiene no puedan ser accedidos o modificados por cualquier persona.

Si tenemos un fichero con la nómina de los empleados en el cual haya un campo que contenga el sueldo, puede ser conveniente que el empleado que opera con los elementos (datos) del fichero no sea capaz de modificar el valor del campo salario.

Si hemos conseguido proteger el campo salario tenemos que estar seguros de que algunas personas no autorizadas no consigan listar los empleados por salario (del que más gana al que menos). Por tanto, no sólo los valores deben estar protegidos, sino las relaciones también.

¿Cómo conseguir esta seguridad?

Los niveles de seguridad para campos se introducen en el diccionario de datos (cuando se especifican los campos y registros de nuestro fichero, al principio). Los códigos de seguridad se introducen en una sección especial del diccionario (por supuesto, pocas personas tendrán acceso a este código de seguridad, si no este invento no valdría para nada).

Campo	Acceso	Actualización
Código	SI	NO
Nombre	SI	NO
Fecha nacimiento	SI	NO
Estado	SI	SI
Sexo	SI	NO
Sueldo	NO	NO
Usuario	**Acceso**	**Actualización**
Empleado	NO	NO
Jefe de personal	SI	NO
Director general	SI	SI

Fig. 6.2.

Debemos evaluar si para nuestro sistema es importante la seguridad.

Si organizamos nuestra agenda mediante una base de datos, no será un factor principal la seguridad. En el otro extremo, aplicaciones de tipo militar necesitarán invertir gran parte del sistema en códigos de seguridad y otras trabas, para que no pueda acceder cualquiera a esa información reservada.

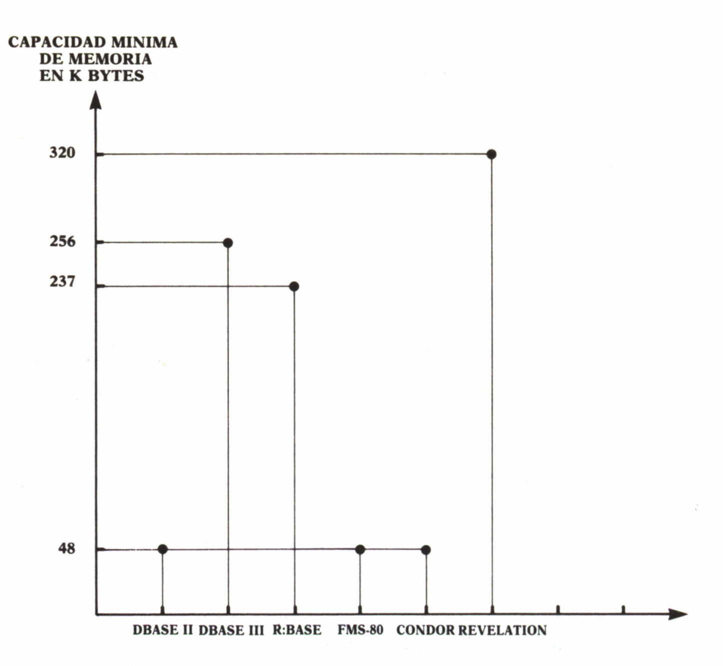

CAPACIDAD MINIMA
DE MEMORIA
EN K BYTES
320
256
237
48
DBASE II
DBASE III
R:BASE
FMS-80
CONDOR
REVELATION

Fig. 6.3.

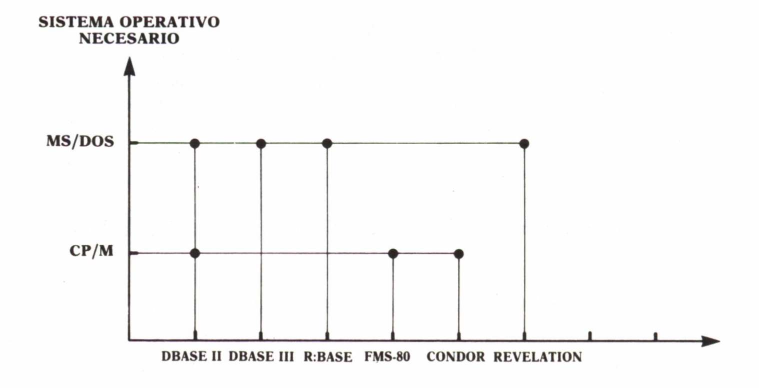

SISTEMA OPERATIVO
NECESARIO
MS/DOS
CP/M
DBASE II
DBASE III
R:BASE
FMS-80
CONDOR
REVELATION

Fig. 6.4.

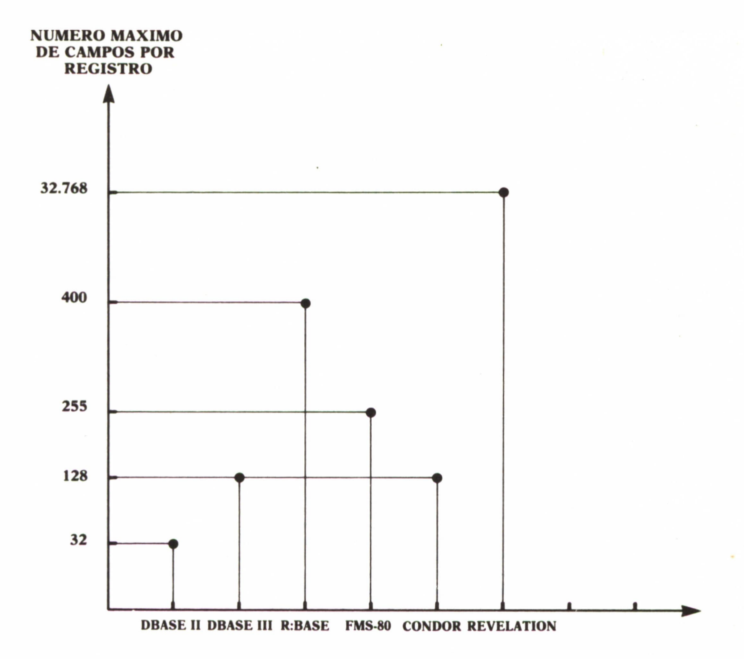

NUMERO MAXIMO
DE CAMPOS POR
REGISTRO
32.768
400
255
128
32
DBASE II
DBASE III
R:BASE
FMS-80
CONDOR
REVELATION

Fig. 6.5.

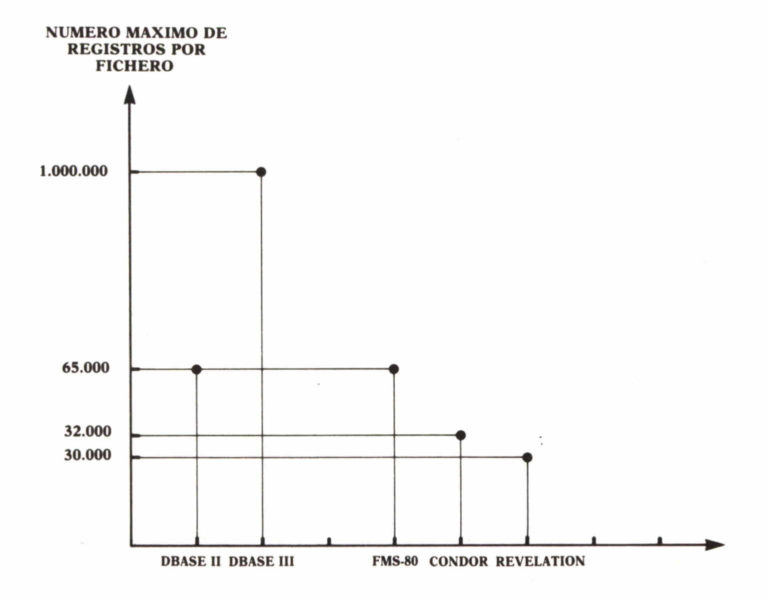

NUMERO MAXIMO DE
REGISTROS POR
FICHERO
1.000.000
65.000
32.000
30.000
DBASE II
DBASE III
FMS-80
CONDOR
REVELATION

Fig. 6.6.

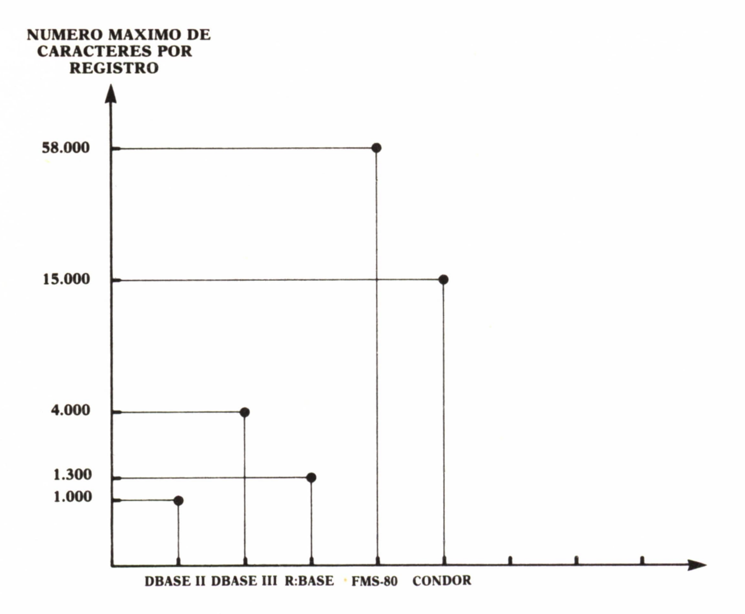

NUMERO MAXIMO DE
CARACTERES POR
REGISTRO
58.000
15.000
4.000
1.300
1.000
DBASE II
DBASE III
R:BASE
FMS-80
CONDOR

Fig. 6.7.

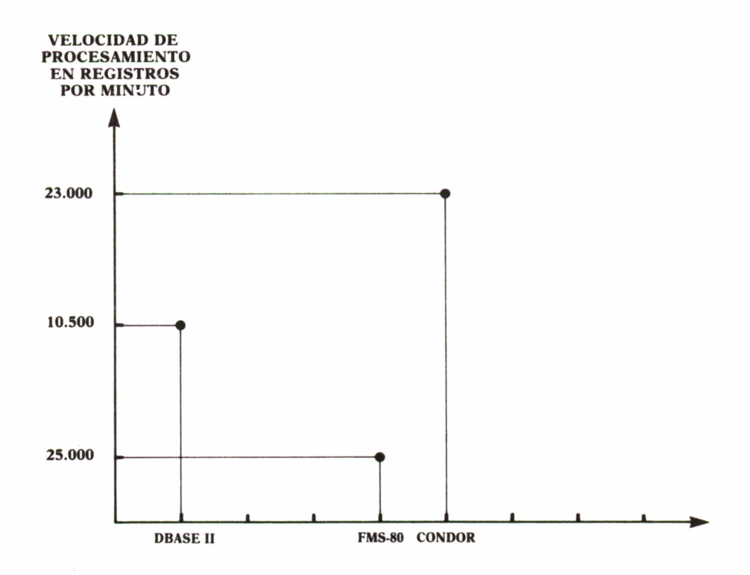

VELOCIDAD DE
PROCESAMIENTO
EN REGISTROS
POR MINUTO
23.000
10.500
25.000
DBASE II
FMS-80
CONDOR

Fig. 6.8.

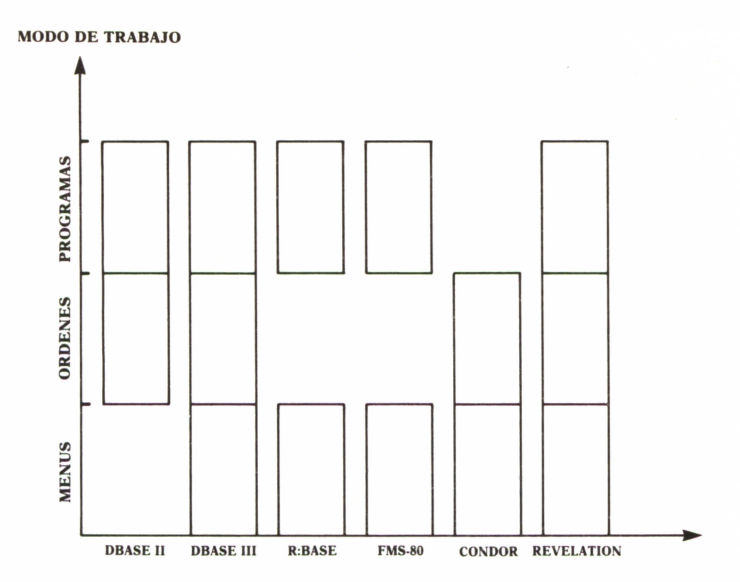


Fig. 6.9.

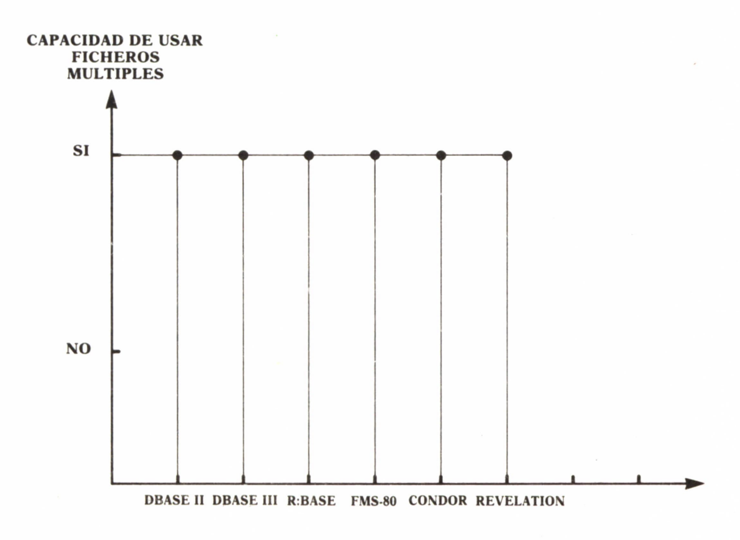


Fig. 6.10.

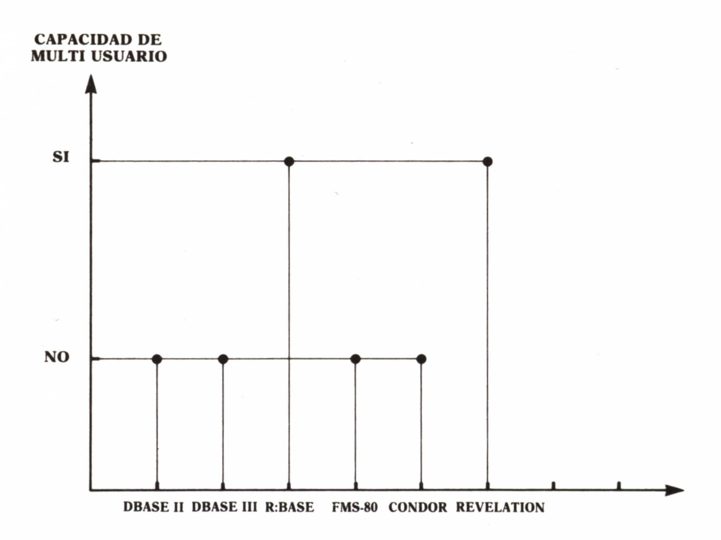
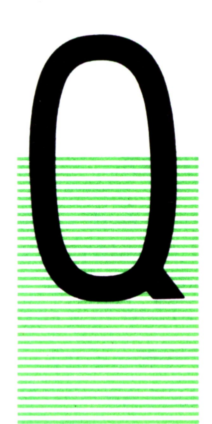

CAPACIDAD DE
MULTI USUARIO
SI
NO
DBASE II
DBASE III
R:BASE
FMS-80
CONDOR
REVELATION

Fig. 6.11.

APENDICE 1

GENERALIDADES



UE es un dato?: Un dato puede ser un número, un nombre o cualquier otro símbolo que nos permite comunicar una información acerca de algo o alguien. El símbolo que utilicemos no varía la información que represente al mismo. Lo importante es el sentido del dato, y no la forma en que está descrito.

Un dato siempre está referido a algo o alguien. Pero este dato no tiene por qué ser una verdad eterna, puede que sólo sea cierto durante un período de tiempo determinado.

¿Cómo se utilizan los datos en un ordenador? Para poder referirnos a un dato necesitamos poder referirnos a él, saber dónde está almacenado en el ordenador. Es decir, para averiguar el valor de un dato necesitamos saber cuál es su dirección en el ordenador. Por ello nos referiremos a los datos mediante su dirección, que podrá ser: **ABSOLUTA** (el byte 8544) o bien **RELATIVA** (se le da un nombre simbólico a la dirección). Aunque al ejecutar el programa para cada dirección relativa debe existir un lugar concreto (dirección absoluta) al que se asimila.

¿Por qué se necesitan las estructuras de datos? Normalmente no trabajamos con datos elementales e inconexos, sino que más bien lo hacemos con conjuntos de datos homogéneos. Por ejemplo, si nuestros datos se refieren a nuestros clientes no nos interesará cada saldo y nombre de cliente como dato individual, ya que obligaría a nuestro programa a manejar tantos datos y direcciones como clientes; por el contrario, sí nos interesa organizar los datos en estructuras conocidas que nos permitan tratar la información de forma homogénea.

¿Qué son las estructuras de datos? Una estructura de datos es un conjunto de datos y un conjunto de funciones de acceso que permiten manipular los datos guardados en algún lugar del ordenador. Las funciones de

acceso son las que definen qué se puede hacer y qué no con los datos almacenados en la estructura. Son esas funciones las que definen el carácter de la estructura.

Tipos de estructuras de datos. Todos los datos que maneja el ordenador están almacenados en su memoria y las intrucciones que los transforman son las instrucciones máquina.

Dado que esas instrucciones son muy elementales, se construyen estructuras más complejas. Existe toda una gradación desde las estructuras del ordenador hasta las más complejas que somos capaces de manejar. Para efectuar esta extensión se necesita, por una parte, definir la nueva estructura (decir qué funciones de acceso posee) y organizar el almacenamiento de esa estructura (en función de otras estructuras ya definidas) e implementar los algoritmos de las funciones de acceso definidas (en función de otras funciones de acceso e instrucciones de programa).

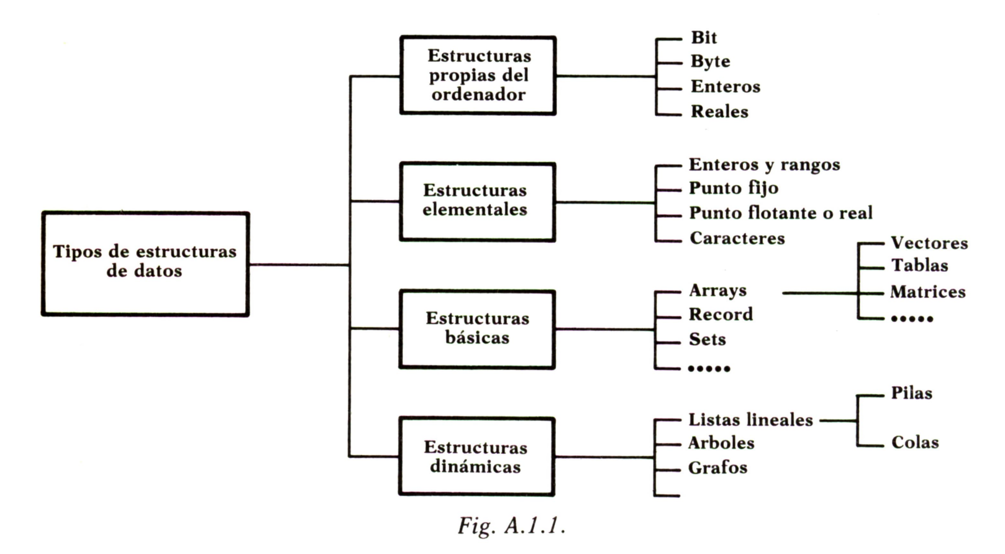


Fig. A.1.1.

Estructuras propias del ordenador. Los datos están almacenados en la memoria y ésta es una estructura de vector que permite acceder a cada elemento a partir de su dirección. Los elementos de la memoria son palabras. Así, el acceso a los datos se debe hacer por palabras dando su dirección absoluta. Estas están formadas por bits. El número de bits que forma la palabra determina la potencia de cálculo simultáneo del procesador. Tamaños habituales de palabra son:

8, 16, 32, 64 bits

Bits. El bit es la unidad elemental de información y puede tomar sólo dos valores: 0 ó 1. Trabajar directamente con bits no sería cómodo, ya que

representan muy poca información. Así, para acceder a un bit se debe acceder a la palabra que lo contiene y dentro de ésta acceder al bit correspondiente.

Byte. Un byte es el número de bits necesarios para repesentar un carácter codificado. El número de bits necesarios depende de la codificación utilizada (ASCII, EBCDIC...).

Enteros. La aritmética del ordenador es la normal, mientras no se alcanza el tope que tiene establecido, por encima del cual no sabe trabajar. La representación interna es binaria, es decir, el número se almacena y opera en forma de número binario un bit de la palabra por cada dígito binario. Dicho tope está en función del número de bits de la palabra. Así, por ejemplo, con 8 bits se pueden utilizar números entre −127 y 128, con 16 bits, entre −32768 y +32767, etc.

Reales. Dado que el ordenador es «finito» y un real puede tener infinitas cifras decimales, lo que se hace es representar los números en «punto flotante» (3,1415 × 10.0...).

Dichos números almacenan por separado las cifras del punto significativas y el orden de magnitud del valor.

Estructuras elementales. Las estructuras lógicas elementales, no necesariamente existentes en el ordenador, son datos estructurados. Estas pueden ser:

ENTEROS (1000, 317)
PUNTO FIJO (18.82, 14.45...)
PUNTO FLOTANTE OREAL (3,15 X 10^2 ...)
CARACTERES («A», «S»,...)
BITS, BOOLEANAS (False, «1»...)
"
"
"
"

Estructuras básicas. Son agrupaciones de estructuras elementales que tienen un número de elementos fijo, que no puede variar con el tiempo. Estas pueden ser:

ARRAYS (rectas, tablas o matrices). Son estructuras de datos que agrupan datos del mismo tipo, de forma que la única diferencia (a nivel de dirección) es el lugar que ocupan.

RECORD (Registro Lógico de Datos). Un récord es una estructura que agrupa datos de distintos tipos (enteros, caracteres, etc.), pero referidos al mismo objeto. (Datos personales de alguien: nombre, apellidos, teléfono...). A cada dato de un récord se le llama campo.

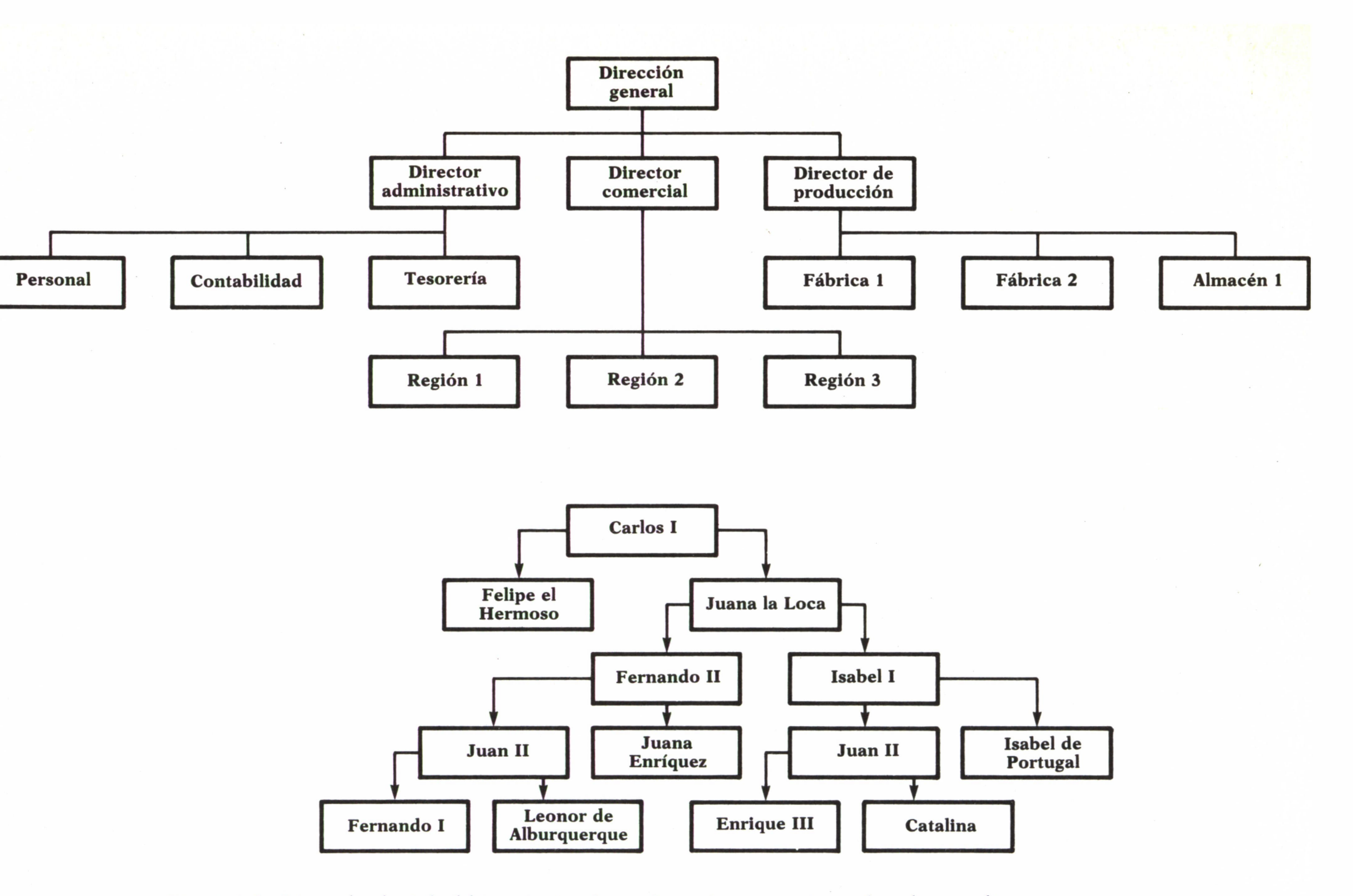


Fig. A.1.2. Ejemplo de árbol binario (cada padre a lo sumo tiene dos descendientes).

CONJUNTOS Estructura que agrupa en un dato los elementos de un conjunto que pertenecen a un tipo determinado. Ejemplo: colores = rojo, verde, azul; meses de verano = julio, agosto, septiembre.

Estructuras dinámicas. Son agrupaciones de estructuras que no tienen un número fijo de elementos. Se utilizan para representar elementos cambiantes en el tiempo. (Ejemplo: una lista de clientes, una lista de expedientes pendientes de ser resueltos).

Listas lineales. Son estructuras dinámicas que sólo necesitan una relación lineal, es decir, un elemento sólo se relaciona con el anterior y el posterior.

Las más frecuentes son:

— Filas: representan una fila real.

— Colas: representan una cola real.

Arboles. Son estructuras jerárquicas en las que un elemento sólo tiene un superior o padre, pero un padre puede tener muchos hijos. Arboles binarios son aquellos en los que un padre puede tener a lo sumo dos hijos.

Grafos. Estructuras en las que un elemento puede estar relacionado con más de dos y sin que exista una relación jerárquica. A cada elemento se le llama nodo y al conjunto red. Son las estructuras más complejas.

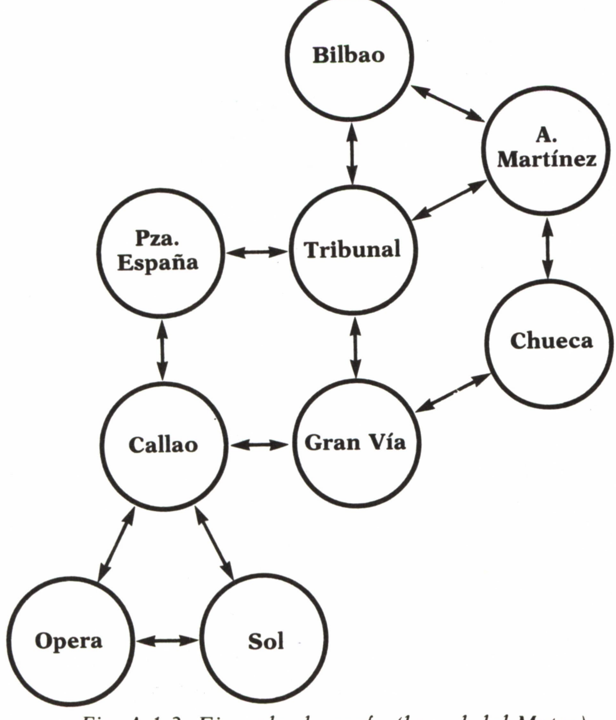


Fig. A.1.3. Ejemplo de grafo (la red del Metro).

MODELOS DE BASES DE DATOS

En la teoría de bases de datos se distinguen tradicionalmente tres modelos de organización de los datos: JERARQUICO, EN RED, RELACIONAL.

Es importante a la hora de adquirir un sistema de gestión de base de datos saber cuál es el modelo de datos que mejor se adapta a la estructura de nuestra organización o a nuestras necesidades de información.

Una base de datos no es más que un intento de modelizar el mundo real para ser representado en soporte informático. Debemos distinguir en un modelo de bases de datos los siguientes elementos:

— *Objetos:* elementos del mundo real de los que queremos tener información como pueden ser clientes, pedidos, publicaciones, etc.

— *Atributos:* características que definen a los objetos.
— *Relaciones:* correspondencias entre los objetos de nuestro modelo.

Según sean las relaciones que se implementen, tendremos un modelo u otro de datos.

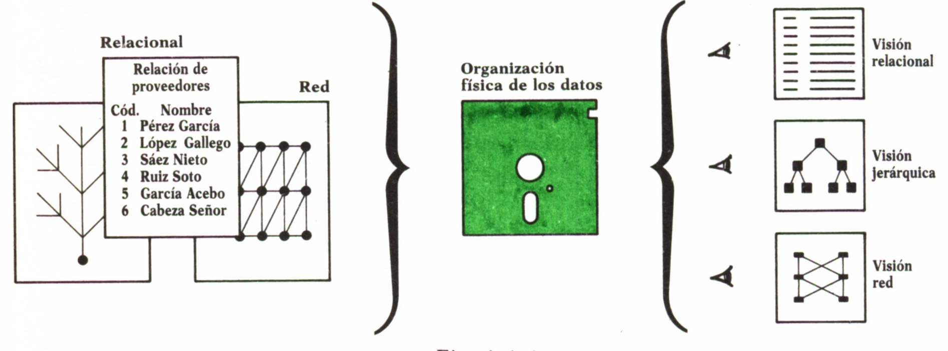


Fig. A.1.4.

En las bases de datos JERARQUICAS la relación existente entre los datos será similar a la que hay entre los nodos de un árbol: la raíz se puede considerar como el nodo más importante jerárquicamente, debajo de ellas se encontrará el nodo terminal del tronco, etc.

El modelo de RED permite establecer una relación mucho más dinámica, ya que cada «nodo» de la red, que será el elemento fundamental, puede estar conectado prácticamente con cualquier otro «nodo».

El último modelo, como su nombre indica, implica trabajar con relaciones; tal es el caso de una sencilla relación de proveedores de una empresa, integrada por un código de identificación y el nombre del proveedor.

De lo anterior se puede deducir que la diferencia fundamental existente entre los tres modelos estriba en la forma en que permiten *«visualizar»* la información. El término VISUALIZAR hace aquí referencia tanto a la forma en la que pueden leer o escribir información en la base de datos los programas de aplicación como a la estructura con la que puede trabajar directamente el usuario sin necesidad de que intervenga más que el propio software del sistema gestor de la base. Por tanto, la diferencia entre 2 modelos de datos no estriba tanto en la forma que adopta el almacenamiento físico de los datos en los soportes de información, sino en su METODO LOGICO DE EXPLOTACION.

A continuación detallaremos un poco más las bases relacionales, por ser el modelo más extendido entre los microordenadores.



EL MODELO RELACIONAL DE BASE DE DATOS

La relación es el elemento fundamental y definitorio de este modelo de base de datos. Intuitivamente, una relación puede asociarse a una matriz en la que cada columna tiene, además de un significado específico, un cierto dominio dentro del que puede tomar valores numéricos o alfanuméricos. Dentro de la matriz, cada fila (usualmente denominada «tupla») representa a un elemento de la relación.

Partiendo de este fundamento, puede definirse a la base de datos relacional como «un conjunto finito de relaciones variables con el tiempo, definidas sobre un conjunto finito de dominios».

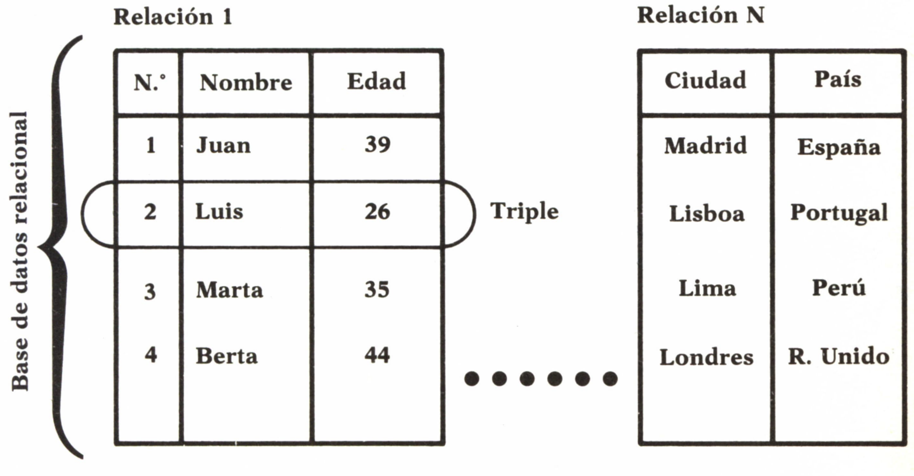


Fig. A.1.5.

La mayor ventaja de este modelo es la facilidad de uso y la claridad de presentación de información conseguida con las tablas. El único problema que puede plantear este modelo estriba en la consecución de un buen diseño en relaciones. Ello supone asociar los datos en tratamiento o relaciones que garanticen una cómoda y eficaz actualización y obtención de los datos.

Para llegar a un buen diseño, puede adoptarse la teoría estudiada y formalizada por Codd (1970). En síntesis, dicha teoría transforma una base de datos en la que las relaciones no deben satisfacer ninguna propiedad especial, en otra base de datos normalizada. Para operar esta transformación se va imponiendo sucesivamente una serie de condiciones. Con ello, se pasa del diseño inicial a SUCESIVOS diseños que responden a la denominación de «primera forma normal», «segunda forma normal» y «tercera forma normal de Codd». Existen hasta una «quinta forma normal», aunque habitualmente es suficiente llegar a que se cumpla la tercera forma normal de Codd para conseguir un modelo suficientemente válido.

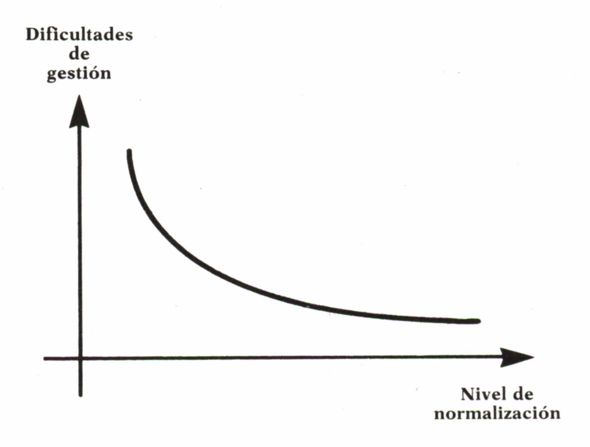


Fig. A.1.6.

A medida que va aumentando el ordinal de normalización, la organización de las relaciones que forman la base resulta más adecuada para su explotación.

De entre las aplicaciones informáticas más prácticas estudiadas y difundidas, destacan por su utilidad las bases de datos. En el mercado existen muchas, pero ¿cuál es la más apropiada para su problema concreto?

El propósito de este libro es responder a esta pregunta. Su lectura le permitirá disponer de los elementos de juicio suficientes para poder no sólo elegir adecuadamente la base de datos más apropiada, sino también conocer sus aplicaciones más interesantes.